On Streaming Disaster Damage Assessment in Social Sensing: A Crowd-driven Dynamic Neural Architecture Searching Approach

Yang Zhang[a], Ruohan Zong[a], Ziyi Kou[b], Lanyu Shang[b], Dong Wang[b,*]

[a]*Department of Computer Science and Engineering*
University of Notre Dame, Notre Dame, IN, USA
[b]*School of Information Sciences*
University of Illinois Urbana-Champaign, Champaign, IL, USA

Abstract

Motivated by the recent advances in Internet and communication techniques and the proliferation of online social media, social sensing has emerged as a new sensing paradigm to obtain timely observations of the physical world from "human sensors". In this study, we focus on an emerging application in social sensing – *streaming disaster damage assessment (DDA)*, which aims to automatically assess the damage severity of affected areas in a disaster event on the fly by leveraging the streaming imagery data about the disaster on social media. In particular, we study a *dynamic optimal neural architecture searching (NAS)* problem in streaming DDA applications. Our goal is to dynamically determine the optimal neural network architecture that accurately estimates the damage severity for each newly arrived image in the stream by leveraging human intelligence from the crowdsourcing systems. The present study is motivated by the observation that the neural network architectures in current DDA solutions are mainly designed by artificial intelligence (AI) experts, which often leads to non-negligible costs and errors given the dynamic nature of the streaming DDA applications and the lack of real-time annotations of the massive social media

*Corresponding author
Email addresses: yzhang42@nd.edu (Yang Zhang), rzong@nd.edu (Ruohan Zong), ziyikou2@illinois.edu (Ziyi Kou), lshang3@illinois.edu (Lanyu Shang), dwang24@illinois.edu (Dong Wang)

data inputs. Two critical technical challenges exist in solving our problem: i) it is non-trivial to dynamically identify the optimal neural network architecture for each image on the fly without knowing its ground-truth label *a priori*; ii) it is challenging to effectively leverage the imperfect crowd intelligence to correctly identify the optimal neural network architecture for each image. To address the above challenges, we developed CD-NAS, a dynamic crowd-AI collaborative NAS framework that carefully explores the human intelligence from crowdsourcing systems to solve the dynamic optimal NAS problem and optimize the performance of streaming DDA applications. The evaluation results from a real-world streaming DDA application show that CD-NAS consistently outperforms the state-of-the-art AI and NAS baselines by achieving the highest disaster damage assessment accuracy while maintaining the lowest computational cost.
Keywords: Crowdsourcing, Social Sensing, Neural Architecture Searching, Disaster Damage Assessment

---

1. Introduction

Social sensing has emerged as a powerful sensing paradigm for collecting observations of the physical world through social media [1, 2]. Examples of social sensing applications include city-wide traffic surveillance using Twitter feeds [3], urban anomaly detection using Foursquare check-ins [4], and community disease outbreak monitoring using Facebook posts [5]. Unlike other infrastructure-based sensing paradigms (e.g., CCTV cameras, remote sensing, wireless sensor networks), social sensing provides a pervasive and scalable solution for obtaining real-time damage information during disaster events [6]. In this paper, we focus on an emerging application in social sensing: *streaming disaster damage assessment (streaming DDA)*[7]. The goal of streaming DDA applications is to automatically assess the damage severity of affected areas in a disaster event *on the fly* by leveraging the streaming imagery data posted on social media. The outputs of streaming DDA applications can be shared with emergency response agencies (e.g., Federal Emergency Management Agency (FEMA), fire

departments) for timely rescue and recovery operations.

Recent advancements in artificial intelligence (AI) have helped in improving the performance of DDA applications [8, 7, 9, 10]. In particular, compared with the traditional DDA solutions that largely rely on intensive manual labeling efforts from disaster specialists [11], the AI-driven DDA solutions significantly reduce the labeling costs while providing a reasonable assessment accuracy [12]. However, current AI-driven DDA solutions often require inputs from experts who are specialists in both AI models and DDA applications to design an appropriate neural network architecture for a particular DDA application. This manual neural network architecture design process is known to be both time-consuming and suboptimal [13]. Figure 1 shows an example where the optimal neural network architecture in a streaming DDA application changes over time. In particular, we observe that the optimal neural network architectures for disaster-related images collected in consecutive timesteps in the same disaster event are different. In such scenarios, it is difficult for AI experts to predict and design an individual optimal neural network architecture for each newly arrived image on the fly. Motivated by the above observations, we study a *dynamic optimal neural architecture searching (NAS)* problem in streaming DDA applications where the goal is to dynamically determine the optimal neural network architecture that accurately estimates the damage severity for each newly arrived image without the inputs from AI experts.

In this study, we develop a *crowd-driven dynamic neural architecture search (CD-NAS)* system to address the above problem by exploring the collective intelligence of both AI and humans. The objective of our CD-NAS design is to leverage human intelligence from crowdsourcing systems to guide the discovery of the optimal neural network architecture for every image in a streaming DDA application. In particular, we observe that human perception is often more reliable and consistent than AI algorithms in terms of identifying the severity of disaster damage from the image (e.g., we can clearly determine the damage severity of images reported in Figure 1). Such human intelligence could possibly help us dynamically identify the optimal neural network architecture in

Figure 1: Changes in Optimal Neural Network Architecture Over Time in Streaming DDA Applications

a streaming DDA application. To obtain accessible and timely human intelligence, we leverage widely adopted open crowdsourcing platforms (e.g., Amazon Mechanical Turk) [14]. We refer to the human intelligence collected from the crowdsourcing platform as *crowd intelligence* in the remainder of this paper. Two important technical challenges exist in designing such a crowd-driven NAS system, which are elaborated below.

Dynamic optimal neural architecture searching. The first challenge lies in the *dynamic* identification of the optimal instance of neural network architecture for each image in the streaming DDA application without knowing its ground truth label *a priori*. In particular, current NAS solutions in AI are mainly designed to identify a single best-performing neural network architecture for a given set of training data [15, 16] and leverage the identified neural network architecture to estimate the damage severity for all testing data. However, such a *one-size-fits-all* neural network architecture could inaccurately estimate the damage severity for a non-negligible portion of images because the optimal neural network architecture often changes over time (as shown in Figure 1).

Recent advancements in dynamic neural networks could potentially be applied to address this issue [17, 18]. However, a major limitation of these solutions is that they still require a large amount of high-quality training labels from the studied disaster event to periodically retrain their models to capture the dynamics of streaming data. However, such a high-quality training dataset is often not available for an *unfolding* disaster in streaming DDA because of the "cold start" problem [19] and the lack of real-time annotations due to cost and resource constraints [20]. Additionally, recent efforts in online deep learning have been made to dynamically update the learned network instances [21, 22]. However, these solutions primarily focus on periodically optimizing the performance of a particular neural network architecture pre-defined by the AI experts on relatively simple tasks (e.g., tweet text classification and handwriting number identification). As a result, their performance may be suboptimal because of the potential bias and constraints of the manual network design process given the excessive damage characteristics and fine-grained details of disaster-related social media images [23]. Therefore, the dynamic identification of the optimal neural network architecture for each incoming imagery data in streaming DDA applications remains a nontrivial question.

Imperfect crowd intelligence-driven NAS. The second challenge lies in leveraging the imperfect crowd intelligence from potentially unreliable crowd workers to facilitate the identification of the optimal neural network architecture in streaming DDA applications. Unlike AI experts who are capable of designing effective neural network architectures, crowd workers are often limited to simplified annotation tasks (e.g., labeling damage severity levels for assigned images). More importantly, unlike the damage severity annotated by disaster specialists, the labels from crowd workers are often imperfect (biased, noisy, and even conflicting responses from different crowd workers) [24]. Additionally, the noise embedded in crowd intelligence can be amplified during the neural network architecture search process, leading to the selection of the poorly performed neural network architecture [15, 25]. Therefore, the key question in our design is how to effectively transfer potentially imperfect crowd knowledge (e.g.,

noisy crowd labels) into an accurate neural network architecture selection for streaming imagery data.

To address the above challenge, we developed *CD-NAS*, a crowd-driven dynamic neural architecture searching approach that carefully explores crowd intelligence to solve the *optimal neural architecture search* problem and optimize the performance of streaming DDA applications. To address the first challenge, we develop a streaming neural network architecture search framework that recursively updates the optimal neural network architecture for each incoming image through a novel recursive maximum likelihood estimation model. To address the second challenge, we designed a novel crowd-AI fusion model that translates imperfect crowd intelligence to effective neural network architecture selection through a robust crowd-AI collaborative network searching process. To the best of our knowledge, CD-NAS is the first *dynamic* crowd-driven NAS approach for solving the streaming DDA problem. We evaluated CD-NAS using a real-world streaming DDA application from a recent disaster event, Typhoon Hagupit. The evaluation results show that our CD-NAS consistently outperforms both state-of-the-art AI and NAS baselines by achieving the highest disaster damage assessment accuracy while maintaining the lowest computational cost under various evaluation scenarios.

A preliminary version of this study was published in [26]. The journal paper is a significant extension of previous work in the following aspects. First, we identify two new intrinsic challenges (i.e., *dynamic optimal neural architecture searching* and *imperfect crowd intelligence-driven NAS*) to solve the dynamic optimal NAS problem and explicitly discuss how our scheme addresses these two challenges (Section 1 and Section 4). Second, we extend the *dynamic optimal architecture searching (DOAS)* module in CD-NAS by developing a dynamic neural network architecture searching scheme that adaptively updates the estimation of the optimal neural network for each image through a recursive estimation framework (Section 4). Third, we extend the evaluation in the conference paper by explicitly studying the performance of all compared schemes with a diversified set of crowdsourcing settings (i.e., different numbers of crowd workers

and crowd query frequencies). The new results demonstrate the effectiveness of our scheme in explicitly leveraging crowd intelligence to guide the discovery of the optimal neural network architecture under different streaming DDA application scenarios (Section 5). Fourth, we add a new study to evaluate the computational cost of all compared schemes (i.e., the average computation time required to estimate the damage severity of an image). This is motivated by the fact that the computational cost is critical in *streaming DDA* applications, especially in the context of massive social media data inputs. The new results demonstrate that our CD-NAS scheme takes orders of magnitude less time to accomplish the streaming DDA task compared with other baselines (Section 5). Fifth, we compare CD-NAS with two additional deep learning and NAS baselines (i.e., *DenseNet* and *MnasNet*) and demonstrate the performance gains achieved by CD-NAS compared with all baselines (Section 5). Sixth, we add a new robustness study to evaluate the robustness of the CD-NAS by varying one key parameter in our design: the size of the sliding window for streaming DDA applications (Section 5). Finally, we extend the related work by adding discussions on recent progress in social sensing and NAS Both of these topics are closely related to the theme of this study (Section 2).

2. Related Work

2.1. Social Sensing

Motivated by the recent advances in Internet and communication techniques (e.g., 4/5G, Internet of Everything (IoE)), as well as the proliferation of online social media (e.g., Twitter and Instagram), social sensing has emerged as a new sensing paradigm to obtain timely observations of the physical world from "human sensors" [27]. Examples of social sensing applications include monitoring real-time traffic conditions in a metro area using mobile crowdsensing to enhance traffic safety [3], obtaining situational awareness in the aftermath of a disaster using online social media for rapid disaster response [28], and detection of infectious disease outbreaks in big cities using location-based crowd track-

ing services to improve public health [5]. Several key challenges exist in the current social sensing applications. Examples include real-time guarantee, data reliability, incentive design, privacy protection, and noise reduction [29, 30, 31]. However, the crowd-driven dynamic optimal NAS problem in streaming DDA applications remains an unsolved challenge in social sensing. In this paper, we address this problem by developing a novel crowd-AI collaborative NAS framework to accurately assess the damage severity of affected areas on the fly using streaming imagery data posted on social media.

2.2. Disaster Damage Assessment

Recent advances in AI and deep learning have been proved remarkably helpful in improving the performance of DDA applications [8, 7, 9, 10, 32, 33]. For example, Li *et al.* developed a deep domain adaptation approach to estimate the damage severity of affected areas using online social media data via adversarial transfer learning [8]. Nguyen *et al.* proposed a deep convolutional network framework for disaster damage assessment of unfolding disaster events for timely disaster response [7]. Kumar *et al.* proposed a deep image classification framework to identify disaster-affected cultural heritage sites from social media imagery data via an end-to-end deep image processing system design [9]. Mouzannar *et al.* developed a deep neural network approach that utilizes both text and image data from social media posts for damage identification via multimodal convolutional neural networks [10]. However, current AI-driven DDA solutions often require extensive inputs from AI experts to design an effective neural network architecture for DDA tasks. Such a manual design process is known to be both error-prone and time-consuming in the presence of massive social data inputs in streaming DDA applications [13]. Efforts on dynamic neural networks in DDA are also relevant to our work [17, 18]. However, two limitations prevent them from being applied to address our problem: i) those methods often require periodical model retraining that often cannot catch up with the large dynamics in our streaming DDA application settings [34]; ii) the performance of these models often drops significantly when they are retrained

using the imperfect crowd labels [35]. In contrast, our CD-NAS framework effectively identifies the optimal neural network architecture for each image without the inputs from AI experts and in the absence of ground-truth labels of newly arrived images.

2.3. Crowd Intelligence

Our work is also related to the growing trend of utilizing pervasive and scalable human intelligence from crowdsourcing systems to solve complex real-world problems [36, 37, 38, 39, 40]. For example, Harris *et al.* leveraged mobile crowdsourcing to detect the defected and deteriorated urban infrastructure for smart city management [37]. Dos Reis *et al.* utilized citizen scientists to segment cancer cells from breast tumors in biomedical research [38]. Wang *et al.* used road traffic information reported by common citizens to monitor real-time traffic congestion in intelligent transportation [40]. However, two fundamental limitations exist in current solutions that fully rely on human intelligence from crowdsourcing systems. First, these approaches may be too labor-intensive and costly compared to our CD-NAS which only requires crowd labels from a small subset of studied images to guide the discovery of the optimal neural network architecture for desirable DDA performance [12]. Second, unlike the professional annotations from disaster specialists, labels from crowd workers can be biased, noisy, and even conflicting because of the lack of sufficient expertise on disaster assessment and response [24]. As a result, the current crowdsourcing solutions could suffer from a non-trivial DDA performance drop by using *only* the imperfect responses from crowd workers. In contrast, our CD-NAS jointly integrates the inputs from crowd workers and AI models into a novel crowd-AI collaborative model that effectively fuses intelligence from both the crowd and AI to address the imperfect crowd response challenge and identify the optimal neural network architecture in DDA applications.

2.4. Neural Architecture Searching

Our work also resembles the NAS technique that is used to automate the neural network design process in many AI-driven real-world applications [15,

16, 25, 41, 42]. For example, Zoph *et al.* developed a scheduled drop path mechanism to enable an effective neural network architecture search for semantic image segmentation [15]. Liu *et al.* proposed a differentiable architecture representation mechanism to effectively refine the neural network architecture during the NAS process in natural language modeling [16]. Tan *et al.* designed a lightweight NAS approach to incorporate model inference latency into the factorized hierarchical searching process for image object detection via multi-objective reinforcement learning [25]. Mo *et al.* proposed a recursive NAS approach to concurrently search for the optimal network architecture on layer and network block levels to improve the NAS performance in keyword spotting on smart devices [41]. To the best of our knowledge, CD-NAS is the first NAS solution that effectively transfers imperfect crowd intelligence to dynamic optimal neural network architecture selections in streaming DDA applications.

3. Problem Description

In this section, we formally define our crowd-driven dynamic NAS problem in streaming DDA applications. We first define a few key terms that will be used in the problem formulation.

Definition 1. ***Disaster-related social media images*** *(X): We define X to represent the disaster-related images posted by common citizens on social media (e.g., Twitter) during a disaster event (as shown in Figure 2), where each posted image captures a specific scene of the studied disaster event.*

Definition 2. ***Social media image stream*** *(S): We define $S = \{X_1, X_2, ..., X_T\}$ as the set of streaming social media images collected during a disaster event, where X_t represents the disaster-related social media image collected from the t^{th} timestep and T is the total number of timesteps in the studied streaming DDA application (e.g., see Figure 1).*

Figure 2: Examples of Disaster-related Social Media Images

Definition 3. ***Damage severity level*** *(L): We define the damage severity level L to represent the severity of the damage captured in a disaster-related social media image. In particular, we define* $L = \{L_1, L_2, ..., L_T\}$ *to represent the damage severity levels for all collected social media images, where* L_t *represents the damage severity level for* X_t*.*

Definition 4. ***Categories of damage severity level*** *(K): Following a similar procedure in [7], the damage severity level in an image can be classified into one of the K pre-defined categories:* $L_t \in \{1, 2, ..., K\}$*. For example, we can consider three categories of damage severity levels (i.e., K =3) that include severe damage, mild damage, no/minor damage as shown in Figure 2.*

Definition 5. ***Neural network architecture search space*** *(N): We define* $N = \{N_1, N_2, ..., N_E\}$ *as an NAS search space that contains a set of E different neural network architecture candidates for streaming DDA tasks, where* N_e *represents a neural network architecture candidate in N (e.g., architecture 1 and 2 in Figure 1). In this study, we leverage the neural network architecture design space (i.e., different configurations of adopting ImageNet-pre-trained convolutional layers for image classification tasks [43]), which is commonly adopted in the current AI-driven DDA solutions [7, 11].*

Definition 6. ***Damage severity estimation from AI*** *(*$\widehat{L^N}$*): We define* $\widehat{L^N}$ *as the damage severity level estimated by different neural network architectures*

in N. In particular, $\widehat{L_t^{N_e}}$ represents the damage severity level estimated by the neural network architecture N_e for the reported image X_t.

Definition 7. ***Dynamic optimal network architecture*** *(N^*): We define N^* as the set of optimal neural network architectures identified by our CD-NAS framework from N for different images in S. In particular, N^{t*} represents the optimal neural network architecture that produces the most accurate damage severity estimation $L_t^{N^{t*}}$ for the image X^t collected at the t^{th} timestep (e.g., N^{t*} is set to be architecture 1 at timestep 1 and architecture 2 at timestep 2 in Figure 1).*

The goal of our crowd-driven dynamic NAS problem is to leverage human intelligence from the crowdsourcing systems to improve the performance of streaming DDA applications. In particular, our goal was to *dynamically* select the optimal neural network architecture for each image. We formally define our problem as follows:

$$\arg\max_{N^{t*}} \Pr(\widehat{L_t^{N^{t*}}} = L_t \mid X), \quad \forall 1 \leq a \leq T \tag{1}$$

This problem is challenging because of the difficulty of transferring the imperfect crowd intelligence to dynamically identify the optimal neural network architecture for streaming social media image data in the absence of ground-truth labels. In this paper, we develop a CD-NAS system to address this problem, which is elaborated in the next section.

4. Solution

In this section, we present the CD-NAS framework to address the dynamic optimal neural architecture search problem in streaming DDA applications. We first present an overview of CD-NAS and then discuss its core modules in detail. Finally, we summarize the CD-NAS framework using pseudocodes.

4.1. Overview of CD-NAS Framework

An overview of the CD-NAS is shown in Figure 3. In particular, it consists of two modules: 1) *crowd-driven network architecture selection (CNAS)* and *dynamic optimized architecture searching (DOAS)*. First, the CNAS module develops a novel crowd-AI integration model to effectively leverage imperfect crowd knowledge to facilitate the discovery of an optimal neural network architecture. Second, the DOAS module designs a dynamic neural network architecture searching scheme that adaptively updates the estimation of the optimal neural network for each image through a recursive estimation framework.

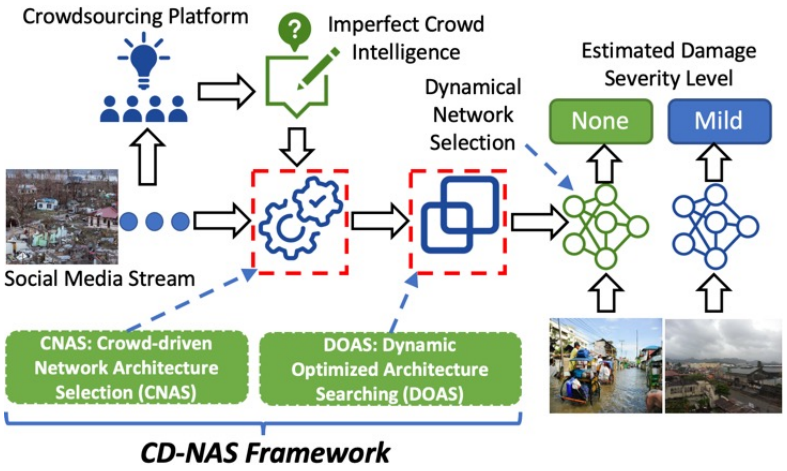


Figure 3: Overview of CD-NAS Framework

4.2. Crowd-driven Network Architecture Selection (CNAS)

In this subsection, we develop a principled crowd-AI integration model to explicitly leverage imperfect crowd intelligence to facilitate the discovery of the optimal neural network architecture in streaming DDA applications. In particular, we first define a key concept that is used in our CNAS module:

Definition 8. ***AI-crowd fusion window*** *(AFW): The AFW is defined as a sliding window for the streaming DDA applications that includes the most recent I images from the social media stream S. In particular, we define AFW =*

$\{X_1, X_2, ..., X_I\}$, where X_i represents the i^{th} image in the sliding window and I is the size of the AFW. We note that I is an application-specific parameter and will study its effect in Section 5.

Similar to the online video applications (e.g., YouTube) that often use a local data buffer to ensure a smooth streaming video service, the AFW here is designed to buffer a set of images in streaming DDA applications for the dynamic neural network architecture search. In particular, we add newly arrived images to the AFW until it is full. Then, we apply the first-in-first-out (FIFO) strategy to replace the oldest image in AFW with the newly arrived image. The optimal neural network architecture for each image in the AFW was identified when the image was evicted from the AFW. Such a design is performed to ensure that our CD-NAS can recursively improve the estimation of the optimal neural network for each image in the AFW.

In our CD-NAS system, we explicitly leverage human intelligence from a crowdsourcing system to guide the discovery of the optimal neural network architecture for each image in AFW. Hence, we further define a few concepts related to crowd intelligence as follows:

Definition 9. ***Crowd query*** *(Q): We define a crowd query as a crowdsourcing task in which our system sends a subset of images in the AFW for the crowd workers to label their damage severity levels. The returned crowd labels are used to search the optimal neural network architecture for each image, which is discussed later in this section when we formally introduce our AI-crowd collaboration model design.*

Definition 10. ***Damage severity labeled by crowd workers*** *($\widehat{L^C}$): In a crowd query Q, each image is labeled by a set of B crowd workers, where C_b is the b^{th} crowd worker in Q. We further define $\widehat{L_i^{C_b}}$ as the damage severity level labeled by a crowd worker C_b for an image X_i.*

Definition 11. ***Crowd query frequency*** *(β): We define β as an application-specific parameter that specifies the frequency to periodically sample the images*

from the DDA data stream for crowd annotations. In Section 5, we study the effect of β.

Unlike labels annotated by domain experts in disaster damage management, the labels from crowd workers are often imperfect (e.g., biased, noisy, and even conflicting with each other) [24]. In particular, the noise embedded in the crowd intelligence can be amplified during the neural network architecture searching process, leading to the selection of the poorly performed neural network architecture. To address the imperfect crowd label challenge, our CNAS module designs a crowd-AI integration model to accurately identify the optimal neural network architecture by leveraging imperfect crowd intelligence. Our design integrates the estimations of different neural network architectures and imperfect crowd responses into a principled estimation framework to estimate the performance of each neural network architecture, N_e, in N. In particular, we observe that every neural network architecture N_e in N and every participating crowd worker C_m in a crowd query Q generate their own estimation of damage severity levels for the images in the AFW. In our CNAS module, we consider both N_e and C_b as data sources with unknown reliability to estimate the variables of unknown ground-truth labels (i.e., disaster-related images with unknown damage severity levels). First, we define an AI-crowd collaboration committee as follows:

Definition 12. ***AI-crowd collaboration committee*** *(M): We define M as a committee that includes both different neural network architectures in N and the crowd workers who participate in the crowd query Q in the streaming DDA application as follows:*

$$M = \{N_1, N_2, ..., N_E, C_1, C_2, ..., C_B\} \tag{2}$$

where N_e represents the e^{th} neural network architecture in N, and C_b represents the b^{th} crowd worker in crowd query Q. In particular, we define M_u as representing the u^{th} committee member in M (i.e., representing either N_e or C_b). In addition, there are a total of $U = E + B$ members in M.

Definition 13. ***Assessment reliability*** *(δ): We define δ_{M_u} to represent the disaster damage assessment reliability of member M_u in M. In particular, δ_{M_u} represents the probability that the estimated damage severity level by M_u is correct.*

Given the above definitions, the goal of our CNAS module is to select the neural network architecture in M with the highest assessment reliability as the optimal neural network architecture in our crowd-driven NAS problem. To that end, we further define $P^{+}_{u,k}$ and $P^{-}_{u,k}$ as the unknown probability that the member M_u estimates the damage severity level of an image to be the k^{th} level and the value other than the k^{th} level given the ground-truth damage severity level of the image is the k^{th} level, respectively. We formally define $P^{+}_{u,k}$ and $P^{-}_{u,k}$ as follows:

$$\begin{aligned} P^{+}_{u,k} &= \Pr(\widehat{L_i^{M_u}} = k | L_i = k) \\ P^{-}_{u,k} &= \sum_{\overline{k} \neq k}^{K} \Pr(\widehat{L_i^{M_u}} = \overline{k} | L_i = k) \end{aligned} \tag{3}$$

where $\widehat{L_t^{M_u}}$ represents the estimated damage severity level by a member M_u in M on an image X_t in AFW. L_i is the ground-truth damage severity level for X_i. Given the above definition, $P^{+}_{u,k}$ and $P^{-}_{u,k}$ are related to the assessment reliability δ_{M_u} using Bayesian theorem as follows:

$$\begin{aligned} P^{+}_{u,k} &= \frac{G_{M_u,k} \times \delta_{M_u}}{d_k} \\ P^{-}_{u,k} &= \frac{G_{M_u,\overline{k}} \times (1 - \delta_{M_u})}{d_k} \end{aligned} \tag{4}$$

where $G_{M_u,k}$ and $G_{M_u,\overline{k}}$ represent the probability that a member M_u estimates the k^{th} damage severity level and values other than the k^{th} level, respectively. d^k represents the prior probability that a randomly selected image belongs to the k^{th} damage severity level. We note that we can learn the assessment reliability score δ_{M_u} if we can obtain the values for the other parameters in the above equation. To that end, we formulate a crowd-AI maximum likelihood estimation (MLE) problem to estimate the *unknown* assessment reliability score δ_{M_u}

for each member M_u in the AI-crowd collaboration committee and *unknown* damage severity level L as follows:

$$\Pr\left((\widehat{L^{M_1}}, \widehat{L^{M_2}}, ..., \widehat{L^{M_U}})|L, (\delta_{M_1}, \delta_{M_2}, ..., \delta_{M_U})\right) \tag{5}$$

where L^{M_u} indicates the damage severity estimated by a neural network architecture L^{N_e} or labeled by a crowd worker L^{C_b} in M.

Given the crowd-AI MLE problem above, we further define the likelihood function $\mathbb{L}(\theta; \omega, Z)$ of our MLE problem as follows:

$$\begin{aligned}\mathbb{L}(\theta; \omega, Z) &= \mathbb{L}(\theta; (\widehat{L^{M_1}}, \widehat{L^{M_2}}, ..., \widehat{L^{M_U}}), L) \\ &= \prod_{i=1}^{I}\Bigg(\sum_{k=1}^{K}(\prod_{u=1}^{U} {P_{u,k}^{+}}^{R_{u,i}^{k}} \times {P_{u,k}^{-}}^{R_{u,i}^{\overline{k}}} \\ &\times (1 - P_{u,k}^{+} - P_{u,k}^{-})^{(1-R_{u,i}^{k}-R_{u,i}^{\overline{k}})} \times d_k \times Z_{i,k})\Bigg)\end{aligned} \tag{6}$$

The above likelihood function represents the likelihood of the observed data ω (i.e., damage severity levels of images in the current AFW estimated by different neural network architectures and crowd workers) and the values of hidden variables Z (i.e., the actual damage severity level of an image) given the estimated parameter θ. Detailed explanations of the parameters in the $\mathbb{L}(\theta; \omega, Z)$ are summarized in Table 1.

4.3. Dynamic Network Architecture Searching (DNAS)

In the previous subsection, we presented our crowd-AI MLE formulation to learn the assessment reliability for each neural network architecture in our AI-crowd collaboration committee. The next question involves adaptively solving the formulated crowd-AI MLE problem to learn the assessment reliability on the fly so that we can dynamically identify the optimal neural network architecture for each image. To that end, we propose a *recursive* expectation maximization (EM) solution to solve the crowd-AI MLE problem. In estimation theory [44], the estimation parameter of an MLE problem can be recursively updated in

Notations	Definitions/Explanations
I	size of AI-crowd fusion window
K	number of damage severity levels
U	number of members in AI-crowd collaboration committee
$R_{u,i}^{k}$	indicator variable that is set to be 1 when a member M_u estimates the damage severity of a given image X_i to be the k^{th} level and is set to be 0 otherwise.
$R_{u,i}^{\overline{k}}$	indicator variable that is set to be 1 when a member M_u estimates the damage severity of a given image X_i to be the value other than k^{th} level and is set to be 0 otherwise.
$Z_{i,k}$	probability that the damage severity of an image X_i in image sliding window to be k^{th} level.
θ	estimation parameter of the model, where $\theta = \{P_{1,k}^{+}, P_{C2,k}^{+}, ..., P_{P^-,U}^{+}; P_{1,K}^{-}, P_{2,K}^{-}, ..., P_{U,k}^{-}; d_k\}$ *for* $k = 1, 2..., K$
ω	observed variable of the model, where $\omega = (\widehat{L^{M_1}}, \widehat{L^{M_2}}, ..., \widehat{L^{M_U}})$
Z	hidden variable of the MLE model, which indicates the damage severity L for each image

Table 1: Notations in Crowd-guided Architecture Searching

consecutive timesteps by considering the streaming data input as follows:

$$\theta_{t+1} = \theta_t + [(t+1) \times I_c(\theta_t)]^{-1} \Phi(X_{t+1}, \theta_t) \tag{7}$$

where θ_t and θ_{t+1} indicate the estimation parameters θ at two consecutive timestep t and $t+1$, respectively. X_{t+1} indicates the nearly arrived image at timestep $t+1$. The estimation parameter θ_{t+1} is used to calculate the updated assessment reliability for each neural network architecture in the AI-crowd collaboration committee using Equation (4). $I_c(\theta_t)^{-1}$ indicates the inverse of the Fisher information of the estimation parameter θ_t at timestep t. $\Phi(X_{t+1}, \theta_t)$ represents the score vector of the observed data (input image X_{t+1}) at timestep $t+1$ given the estimation parameter θ_t from the last timestep t. The key idea of the above streaming formulation is to provide a dynamic solution to recursively update the estimation parameter θ on the fly.

To obtain the Fisher information $I_c(\theta_t)$ and score vector $\Phi(X_{t+1}, \theta_t)$, we first derive the log function of $\mathbb{L}(\theta; \omega, Z)$ by assuming that the correctness of the hidden variable ($Z_{t,k}$) can be correctly estimated when the number of members in the AI-crowd collaboration committee is sufficient. In particular, we can derive the log-likelihood function $\log\mathbb{L}(\theta; \omega, Z)$ as:

$$\begin{aligned}\log\mathbb{L}(\theta;\omega,Z) = \mathbb{L}(\theta;(\widehat{L^{M_1}},\widehat{L^{M_2}},...,\widehat{L^{M_U}}),L) = \\ \sum_{i=1}^{I}\Bigg(\sum_{k=1}^{K}\big(\sum_{u=1}^{U}R_{u,i}^{k}\times\log P_{u,k}^{+}+R_{u,i}^{\overline{k}}\times\log P_{u,k}^{-}+ \\ (1-R_{u,i}^{k}-R_{u,i}^{\overline{k}})\times\log(1-P_{u,k}^{+}-P_{u,k}^{-})+d_k+Z_{i,k}\big)\Bigg)\end{aligned} \tag{8}$$

Given the log-likelihood function $\log\mathbb{L}(\theta; \omega, Z)$, we can derive the inverse of the Fisher information $I_c(\theta_t)^{-1}$ for our problem as follows:

$$I_c(\theta_t)^{-1}_{u,v} = \begin{cases} \frac{{P_{u,k}^{+}}^t\times(1-{P_{u,k}^{+}}^t-{P_{u,k}^{-}}^t)}{d_k\times I\times(1-{P_{u,k}^{-}}^t)}, u=v\in[1,U] \\ \frac{{P_{u,k}^{-}}^t\times(1-{P_{u,k}^{+}}^t-{P_{u,k}^{-}}^t)}{d_k\times I\times(1-{P_{u,k}^{+}}^t)}, u=v\in(U,2U] \\ 0, u\neq v \end{cases}$$

In addition, we can also derive the score vector $\Phi(M_{t+1}, \theta_t)$ from $\log\mathbb{L}(\theta; \omega, Z)$ as follows:

$$\Phi(M_{t+1},\theta_t)_{u,v} = \begin{cases} \sum_{i=1}^{I}Z_{i,k}^{t+1}\times\left(\frac{R_{u,i}^{k}}{{P_{u,k}^{+}}^t}+\frac{1-R_{u,i}^{k}-R_{u,i}^{\overline{k}}}{1-{P_{u,k}^{+}}^t-{P_{u,k}^{-}}^t}\right), u=v\in[1,U] \\ \sum_{i=1}^{I}Z_{i,k}^{t+1}\times\left(\frac{R_{u,i}^{\overline{k}}}{{P_{u,k}^{-}}^t}+\frac{1-R_{u,i}^{k}-R_{u,i}^{\overline{k}}}{1-{P_{u,k}^{+}}^t-{P_{u,k}^{-}}^t}\right), u=v\in(U,2U] \\ 0, u\neq v \end{cases} \tag{9}$$

Finally, we can plug in $I_c(\theta_t)^{-1}$ and $\Phi(M_{t+1}, \theta_t)$ into Equation (7) to obtain the recursive formula to update the estimation parameters θ (i.e., $P_{u,k}^{+}$ and $P_{u,k}^{-}$) as follows:

$$
\begin{aligned}
{P_{u,k}^{+}}^{t+1} &= {P_{u,k}^{+}}^{t} + \frac{1}{I \times d_k \times (1 - {P_{u,k}^{-}}^{t}) \times (t+1)} \times \\
&\left(\sum_{i \in {\Delta_u^k}^{t+1}} Z_{i,k}^{t+1} \times (1 - {P_{u,k}^{+}}^{t} - {P_{u,k}^{-}}^{t}) - \sum_{i \in {\Delta_u^0}^{t+1}} Z_{i,k}^{t+1} \times {P_{u,k}^{+}}^{t} \right) \\
{P_{u,k}^{-}}^{t+1} &= {P_{u,k}^{-}}^{t} + \frac{1}{I \times d_k \times (1 - {P_{u,k}^{+}}^{t}) \times (t+1)} \times \\
&\left(\sum_{i \in {\Delta_u^{\overline{k}}}^{t+1}} Z_{i,k}^{t+1} \times (1 - {P_{u,k}^{+}}^{t} - {P_{u,k}^{-}}^{t}) - \sum_{i \in {\Delta_u^0}^{t+1}} Z_{i,k}^{t+1} \times {P_{u,k}^{-}}^{t} \right)
\end{aligned}
\tag{10}
$$

where ${\Delta_u^k}^{t+1}$ and ${\Delta_u^{\overline{k}}}^{t+1}$ indicate the set of images from the current AFW The member M_u estimates the damage severity as the k^{th} level and value other than the k^{th} level. ${\Delta_u^0}^{t+1}$ indicates the set of images that the member M_u does not make any estimation of the damage severity level (e.g., the images that are not selected for the crowd query).

Given the above equation, we can clearly observe that the estimation of the estimation parameter ${P_{u,k}^{+}}^{t+1}$ and ${P_{u,k}^{-}}^{t+1}$ (which is used to derive the assessment reliability for each member M_u in the AI-crowd collaboration committee) at the current timestep $t+1$ can be computed from their values ${P_{u,k}^{+}}^{t}$ and ${P_{u,k}^{-}}^{t}$ from the previous timestep t and the observed data in the new timestep $t+1$ (i.e., ${\Delta_u^k}^{t+1}$, ${\Delta_u^{\overline{k}}}^{t+1}$, and ${\Delta_u^0}^{t+1}$). In addition, we observe that $Z_{i,k}^{t+1}$ is unknown and can be estimated by its approximation $\hat{Z}_{i,k}^{t+1}$ as follows:

$$
Z_{i,k}^{t+1} \approx \hat{Z}_{i,k}^{t+1} = \frac{W_{n,k}^{t+1} \times d_k}{\sum_{k=1}^{K} W_{n,k}^{t+1} \times d_k} \tag{11}
$$

where $W_{n,k}^{t+1}$ can be computed as follows:

$$
\begin{aligned}
W_{n,k}^{t+1} = \prod_{i=1}^{A} \Bigg(& \Big(\frac{Q_{u,k}^{t+1}}{Q_{u,k}^{t}} \times {P_{u,k}^{+}}^{t}\Big)^{R_{u,i}^{k}} \times \Big(\frac{Q_{u,\overline{k}}^{t+1}}{Q_{u,\overline{k}}^{t}} \times {P_{u,k}^{-}}^{t}\Big)^{R_{u,i}^{\overline{k}}} \\
& \times \Big(1 - \frac{Q_{u,k}^{t+1}}{Q_{u,k}^{t}} \times {P_{u,k}^{+}}^{t} - \frac{Q_{u,\overline{k}}^{t+1}}{Q_{u,\overline{k}}^{t}} \times {P_{u,k}^{-}}^{t}\Big)^{(1 - R_{u,i}^{k} - R_{u,i}^{\overline{k}})} \Bigg)
\end{aligned}
\tag{12}
$$

In summary, the above recursive approach provides a dynamic solution for learning the estimation parameter θ of the crowd-AI MLE problem *on the fly* at

each timestep using the estimation from the previous timestep and the images from the current image sliding window. Finally, we can derive the assessment reliability δ_{M_n} for each member M_u in M dynamically by plugging the updated θ_{t+1} to Equation (4) at each timestep. After obtaining the assessment reliability score for each neural network architecture, we select the neural network architecture with the highest assessment reliability score as the optimal neural network architecture N^* for the image that is about to be evicted from the AFW as follows:

$$\begin{aligned} &\arg\max_{M_u} {\delta_{M_u}}^{t+1}, \text{ where } M_u \in \{N_1, N_2, ..., N_E\} \\ &\text{set } M_u \text{ as } N^{i*} \text{ for } {X_i}^{t+1} \end{aligned} \tag{13}$$

where ${X_i}^{t+1}$ represents the image that is about to be evicted from the AFW at timestep $t+1$. ${\delta_{M_u}}^{t+1}$ represents the updated assessment reliability score at timestep $t+1$. In addition, the estimated damage severity $\widehat{L^{N^{i*}}}$ from the optimal neural network architecture N^{i*} was taken as the final output of our CD-NAS framework for image ${X_i}^{t+1}$.

Finally, we summarize the CD-NAS framework in Algorithm 1. The inputs to the CD-NAS are the set of streaming social media images X_t. The outputs are the dynamically identified optimal neural network architecture N^{t*} and the estimated damage severity level $\widehat{L^{N^{t*}}}$ generated by N^{t*} for each X_t.

5. Evaluation

In this section, we evaluate the performance of the CD-NAS framework using real-world streaming DDA applications from a real world disaster event. The results show that CD-NAS consistently outperforms the state-of-the-art AI and NAS baselines in terms of both damage assessment accuracy and computational cost under various application scenarios.

Algorithm 1 CD-NAS Framework Summary

initialize each N_e in N (Definition 5)
for each incoming X_t (timestep t) **do**
  obtain $\widehat{L_t^{N_e}}$ for each N_e
  if t is a crowd query timestep based on frequency β **then**
    add X_t to Q (Definition 9)
    obtain $\widehat{L_t^{C_b}}$ from Q
  end if
  if AFW is not full **then**
    add X_t to AFW
  else
    calculate $Z_{i,k}$ using Equation (11)
    calculate $P_{u,k}^{+}$ and $P_{u,k}^{-}$ using Equation (10)
    derive δ_{M_u} using Equation (4)
    for each N_e in M **do**
      select M_u with top ranked δ_{M_u} (Equation (13))
    end for
    set M_u as N^{i*} for X_i (X_i to be evicted from AFW)
    obtain $\widehat{L^{N^{i*}}}$ using N^{i*} for X_i
    replace X_i with X_t in AFW
  end if
end for
output $\widehat{L^{N^{t*}}}$ for each X_t

5.1. Dataset and Crowdsourcing Platform

Disaster Damage Assessment Dataset: In our evaluation, we used a real-world dataset on disaster damage assessment collected by [7] [1]. In particular, the dataset consists of social media images collected over the course of Typhoon Hagupit in Philippines (2014). The collected social media images have diversified damage characteristics (e.g., flooding damage, buildings and infrastructure damage, and vehicle damage) as shown in Figure 1. In the dataset, the ground-truth damage severity level of each social media image was manually classified by domain experts into three categories (i.e., severe damage, mild damage, and no/minor damage). In particular, the distributions of different damage severity levels in our dataset were as follows: *severe damage*: 11.2%; *mild damage*: 42.2%; and *no damage*: 46.6%. We keep the ratio of training to testing data as 3:1, the same as in [7]. The training dataset was used to train all the compared AI models for disaster damage assessment.

Amazon Mechanical Turk Platform: To obtain the crowd intelligence, we utilize Amazon Mechanical Turk (AMT) [2], one of the largest crowdsourcing platforms that provides a large number of 24/7 freelance crowd workers to complete assigned tasks with reasonable incentives. In each crowdsourcing task, we ask the crowd workers to label the damage severity level of the image in the query. To ensure the crowd label quality, we select the crowd workers who have an overall task approval rate greater than 95% and have completed at least 1000 approved tasks to participate in our crowdsourcing tasks. We paid $0.20 for each worker per image in our experiment. In our evaluation, we study a diversified set of crowd query settings to create a challenging evaluation scenario for our CD-NAS framework. In particular, we vary the number of participating crowd workers who respond to each queried image (Definition 10) from 3 to 5 and vary the crowd query frequency β (Definition 11) from 1/5 to 1/3.

[1] `https://crisisnlp.qcri.org/`

[2] `https://www.mturk.com/`

5.2. Baselines and Experiment Settings

We compared CD-NAS with a set of representative deep neural network (DNN) and neural architecture searching (NAS) baselines in streaming DDA applications.

- **DNN Baselines:**
    1. **InceptationNet** [45]: a popular deep learning model that accelerates the learning process of the DDA task through a convolution factorization mechanism.
    2. **DenseNet** [46]: a widely used deep neural network approach that establishes dense connections among different network layers to boost the DDA accuracy.
    3. **VGG** [11]: A representative deep convolutional network framework that utilizes recursive deep convolutional operations to ensure the sufficient network depth for a desirable DDA performance.

- **NAS Baselines:**
    1. **NashNetLarge/Mobile** [15]: A state-of-the-art NAS approach that effectively refines the neural network architecture by introducing a scheduled drop path mechanism. In addition to the standard version of NashNet (*NashNetLarge*), we also consider the mobile version of NashNet (*NashNetMobile*) which achieves a better trade-off between the NAS performance and computational efficiency in streaming DDA applications.
    2. **Darts** [16]: a representative NAS framework that introduces a differentiable architecture representation to ensure an effective NAS process.
    3. **MansNet** [25]: A lightweight NAS approach to incorporate model inference latency into the factorized hierarchical architecture searching process via multi-objective reinforcement learning.

To ensure a fair comparison, the inputs to all compared schemes were set to be the same, which included: 1) the input social media images, 2) the ground truth labels of images in the training dataset, and 3) the labeled images from crowd workers. In particular, we retrained all compared baselines using the labels returned by the crowd query to ensure a fair comparison. In addition, we also consider the *random* baseline, which estimates the damage severity for each image by randomly selecting a damage severity level from the possible categories. In our system, we implemented our CD-NAS model using Tensorflow 2.0 [3], and trained our model using the NVIDIA Quadro RTX 6000 GPU. In our experiment, all hyperparameters were optimized using the Adam optimizer [47]. In particular, we set the learning rate to be 10^{-6}. We also set the batch size to be 20, and the model was trained over 300 epochs.

To evaluate the performance of all compared schemes, we adopted three metrics that are widely used to evaluate the performance of multi-class image classification tasks in image processing: 1) *F1-score*, 2) *Cohen's kappa Score ($\mathcal{K}$-Score)* [48], and *Matthews correlation coefficient (MCC)* [49]. We use $\mathcal{K}$-Score and MCC in our evaluation because we have an imbalanced dataset, and these two metrics have been proven to be reliable for imbalanced data [50]. Higher F1-score, $\mathcal{K}$-Score, and MCC indicate better performance.

5.3. Evaluation Results

5.3.1. DDA Classification Accuracy with Different Crowdsourcing Settings

In the first set of experiments, we studied the performance of all the compared schemes with different crowdsourcing settings. First, we vary the crowd query frequency β (Definition 11) from 1/5 to 1/3 for all compared schemes (e.g., we periodically send every one out of three images in the data stream when β is 1/3 in crowd query) while fixing the number of participating crowd workers B (Definition 10) to be 3. Second, we change the number of participating crowd workers B in the crowd query from three to five while fixing the

[3] `https://www.tensorflow.org/`

crowd query frequency to be 1/3. We set the size of the AI-crowd fusion window AFW (Definition 8) to be 40. The evaluation results are presented in Tables 2 and 3. We observed that our CD-NAS consistently outperformed all the compared baselines in all experimental settings. For example, the performance gain of CD-NAS compared to the best-performing baseline (i.e., DenseNet) when the crowd query frequency $\beta = 1/3$ and $B = 3$ on F1-Score, $\mathcal{K}$-Score, and MCC are 5.76%, 7.48%, and 6.00%, respectively. The performance gains of our scheme mainly come from the fact that it adaptively transfers the imperfect crowd intelligence to the optimal neural network selection for each image through the dynamic crowd-AI MLE design. In addition, we further evaluated the performance of our CD-NAS on additional settings of the two experimental variables (i.e., crowd query frequency β and the crowd worker numbers B). We also compared the performance of the CD-NAS with the best-performing baselines from the different categories (i.e., DenseNet for DNN baselines in Table 2 and Table 3, NasNetMobile for NAS baselines in Table 2 and NasNetLarge for NAS baselines in Table 3). The results are shown in Figure 4 and Figure 5, respectively. We observed that CD-NAS consistently outperformed the best-performing baselines on different evaluation metrics for all evaluation settings. Such evaluation results demonstrate the effectiveness of our scheme in leveraging the imperfect crowd knowledge to dynamically identify the optimal neural network architecture for each newly arrived image to provide accurate DDA results across different experimental variable settings.

5.3.2. Computational Efficiency

In the second set of experiments, we compared the computational cost of all the compared schemes (except the trivial *random* baseline) in the studied streaming DDA application. We define the computational cost as the average computational time required to estimate the damage severity of an image. To ensure a fair comparison, we evaluated all schemes using the same NVIDIA Quadro RTX 6000 GPU. The evaluation results are presented in Tables 4 and 5, respectively. We observe that our CD-NAS scheme takes orders of magnitude

Table 2: DDA classification accuracy Comparisons (Varying Crowd Query Frequency)

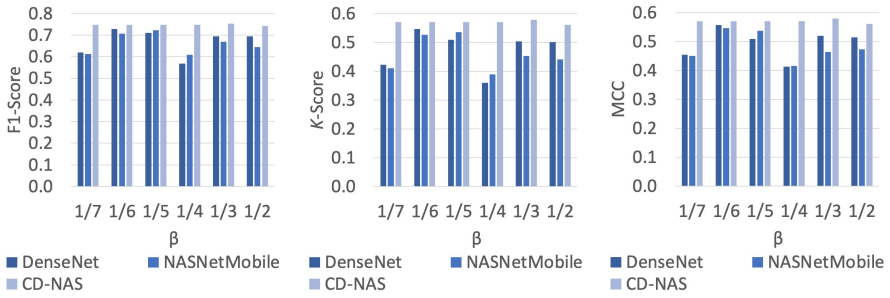
Category	Algorithm	$\beta = 1/5$			$\beta = 1/4$			$\beta = 1/3$		
		F1-Score	$\mathcal{K}$-Score	MCC	F1-Score	$\mathcal{K}$-Score	MCC	F1-Score	$\mathcal{K}$-Score	MCC
Random	Random	0.3291	0.0109	0.0118	0.3664	0.0681	0.0738	0.3416	0.0164	0.0175
DNN	InceptionNet	0.6349	0.3833	0.3834	0.6054	0.3646	0.3857	0.6849	0.4785	0.4871
	DenseNet	0.7097	0.5092	0.5094	0.5684	0.3593	0.4131	0.6949	0.5039	0.5192
	VGG	0.6888	0.4739	0.4757	0.5670	0.3386	0.3711	0.6493	0.4250	0.4319
NAS	NASNetLarge	0.6721	0.4473	0.4478	0.5573	0.3148	0.3410	0.6916	0.4904	0.5020
	NASNetMobile	0.7231	0.5356	0.5367	0.6099	0.3884	0.4150	0.6496	0.4528	0.4637
	DARTS	0.5907	0.3208	0.3225	0.5596	0.2825	0.2984	0.6450	0.3940	0.3988
	MnasNet	0.5600	0.2569	0.2596	0.6331	0.3705	0.3758	0.6183	0.3625	0.3933
Ours	**CD-NAS**	**0.7471**	**0.5696**	**0.5701**	**0.7471**	**0.5696**	**0.5701**	**0.7525**	**0.5787**	**0.5792**

Table 3: DDA classification accuracy Comparisons (Varying Number of Crowd Workers)

Category	Algorithm	B = 3			B = 4			B = 5		
		F1-Score	$\mathcal{K}$-Score	MCC	F1-Score	$\mathcal{K}$-Score	MCC	F1-Score	$\mathcal{K}$-Score	MCC
Random	Random	0.3416	0.0164	0.0175	0.3614	0.0088	0.0090	0.3540	0.0314	0.0334
DNN	InceptionNet	0.6849	0.4785	0.4871	0.6819	0.4667	0.4688	0.6917	0.4893	0.4980
	DenseNet	0.6949	0.5039	0.5192	0.6861	0.4842	0.4939	0.7305	0.5541	0.5604
	VGG	0.6493	0.4250	0.4319	0.6472	0.4174	0.4217	0.6913	0.4919	0.4975
NAS	NASNetLarge	0.6916	0.4904	0.5020	0.6945	0.4907	0.4943	0.7170	0.5346	0.5464
	NASNetMobile	0.6496	0.4528	0.4637	0.6847	0.4767	0.4799	0.6904	0.4894	0.5070
	DARTS	0.6450	0.3940	0.3988	0.6597	0.4233	0.4239	0.6733	0.4410	0.4465
	MnasNet	0.6183	0.3625	0.3933	0.6564	0.4169	0.4214	0.5919	0.3287	0.3613
Ours	**CD-NAS**	**0.7525**	**0.5787**	**0.5792**	**0.7525**	**0.5787**	**0.5792**	**0.7579**	**0.5878**	**0.5883**

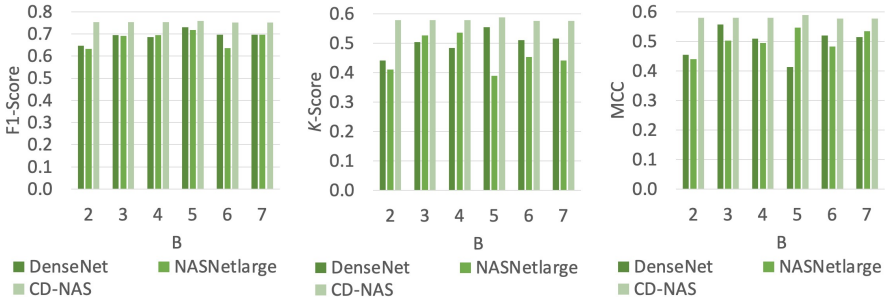
less time to accomplish the DDA task compared to other baselines under different evaluation settings. This is because the compared baselines require additional computational time to retrain their models to capture the dynamics of the streaming data by leveraging the labels from crowd workers. In contrast, our CD-NAS designs a recursive expectation maximization solution that estimates the assessment reliability score of each neural network architecture on the fly without requiring any additional network retraining. In addition, we evaluated

the computational cost of our CD-NAS for additional crowdsourcing settings. Similar to the performance comparison in Section 5.3.1, we compare the performance of the CD-NAS with the best-performing baselines from each category in Tables 2 and 3. The results are shown in Figure 6 and Figure 7. We observe that our CD-NAS achieves a clear performance gain compared to the best-performing baselines in all different settings, which further demonstrates the effectiveness of the dynamic neural network architecture searching scheme in maintaining the best DDA performance while maintaining the lowest computational time cost.



(a) F1-Score (b) $\mathcal{K}$-Score (c) MCC

Figure 4: Performance Comparisons between CD-NAS and Best-performing Baselines (Varying Crowd Query Frequency)



(a) F1-Score (b) $\mathcal{K}$-Score (c) MCC

Figure 5: Performance Comparisons between CD-NAS and Best-performing Baselines (Varying Number of Crowd Workers)

Table 4: Computational Time Comparisons (Seconds) - Varying Crowd Query Frequency

Algorithm	β=1/5	β=1/4	β=1/3
InceptionNet	1.2542	1.4427	1.8782
DenseNet	1.2891	1.5389	1.9914
VGG	1.1466	1.3928	1.7735
NASNetMobile	1.3917	1.5896	2.0130
NASNetLarge	1.5198	1.7942	2.2076
DARTS	0.3067	0.3329	0.3918
MnasNet	0.7937	1.0152	1.1192
CD-NAS	**0.0195**	**0.0197**	**0.0203**

Table 5: Computational Time Comparisons (Seconds) - Varying Number of Crowd Workers

Algorithm	B=3	B=4	B=5
InceptionNet	1.8782	1.8773	1.8790
DenseNet	1.9914	1.9842	1.9923
VGG	1.7735	1.7768	1.7793
NASNetMobile	2.0130	2.0142	2.0121
NASNetLarge	2.2076	2.2084	2.2063
DARTS	0.3918	0.3923	0.3945
MnasNet	1.1192	1.1142	1.1154
CD-NAS	**0.0203**	**0.0198**	**0.0201**

5.3.3. Robustness of CD-NAS Framework

In the third set of experiments, we study the robustness of the CD-NAS by varying one key parameter in our design, that is, the size I of the AI-crowd fusion window AFW (Definition 8). The evaluation results are presented in Figure 8. Given the space limit, we only present the results of one representative crowdsourcing setting (i.e., $B = 3$ and $\beta = 1/3$). The results for the other scenarios are similar. We observe that the performance of CD-NAS is stable as the size of the AFW changes, which demonstrates the robustness of CD-NAS over the key parameter in our model design. The robustness study in Figure 8

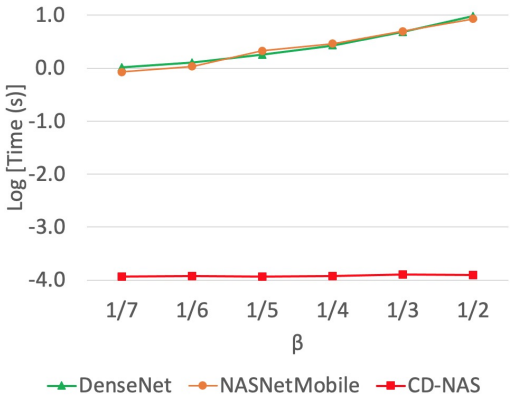


Figure 6: Computational Time Comparisons (Seconds) between CD-NAS and Best-performing Baselines - Varying Crowd Query Frequency

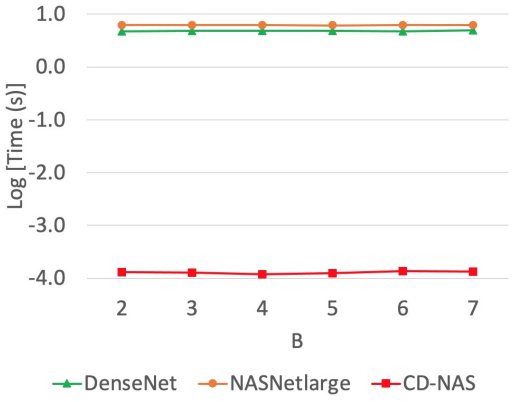


Figure 7: Computational Time Comparisons (Seconds) between CD-NAS and Best-performing Baselines - Varying Number of Crowd Workers

demonstrates that our CD-NAS can achieve consistent DDA performance over a reasonable range of different AFW sizes (i.e., between 25 and 55). The results provide a window for users of our CD-NAS scheme to select the AFW size to achieve a desirable DDA performance. In addition, we also note that the CD-NAS buffers very few images in AFW when its size is too small, which often leads to suboptimal classification results. On the other hand, CD-NAS can buffer too many images in AFW when its size is too large, which often leads to a significantly reduced computation time. The actual selection of the AFW

size will largely depend on the tradeoff between the classification accuracy and response time of the CD-NAS scheme that the users would like to achieve in a particular DDA application.

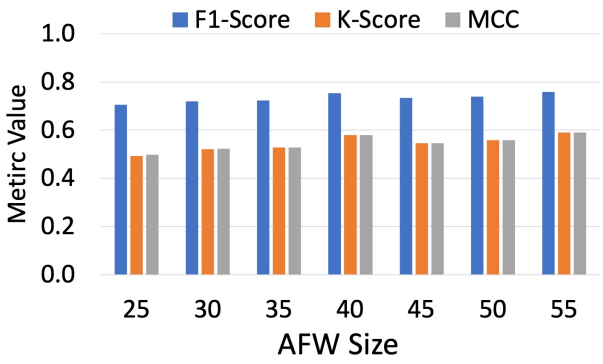


Figure 8: Robustness of CD-NAS Framework

5.3.4. Convergence of CD-NAS Framework

In the last set of experiments, we study the convergence of our CD-NAS by plotting the performance of CD-NAS over different timesteps in the social media image stream (Definition 1). The results are presented in Figure 9. Similar to the robustness study, we only show the performance for one representative setting (i.e., $B = 3$ and $\beta = 1/3$) because of the space limit. The results for the other scenarios are similar. Please note that we show the performance of CD-NAS from the 20^{th} timestep because our CD-NAS needs to explore the imagery data at the first few timesteps to overcome the cold start problem of the recursive EM algorithm. We observe that our CD-NAS can quickly boost the assessment performance and remain stable afterward, suggesting its effectiveness in recursively learning the optimal neural network architecture in the studied application.

6. Conclusion

We presented a CD-NAS framework to address a crowd-driven dynamic NAS problem and improve the QoS of streaming DDA applications. Our solution is

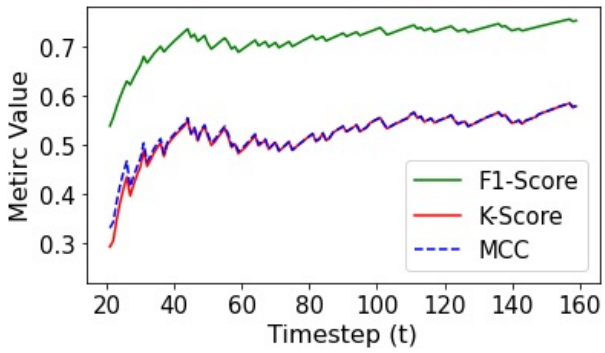


Figure 9: Convergence of CD-NAS Framework

inspired by interdisciplinary techniques such as AI, crowdsourcing, and estimation theory. Our results on a real-world streaming DDA application showed that CD-NAS outperforms AI and NAS baselines in terms of both damage assessment accuracy and computational cost. We believe that CD-NAS will provide useful insights to explore the collective power of AI and crowd intelligence in a rich set of AI-driven streaming applications (e.g., disaster response, truth discovery, intelligent transportation).

Acknowledgment

This research is supported in part by the National Science Foundation under Grant No. CHE-2105005, IIS-2008228, CNS-1845639, CNS-1831669, Army Research Office under Grant W911NF-17-1-0409. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] D. Wang, B. K. Szymanski, T. Abdelzaher, H. Ji, L. Kaplan, The age of social sensing, Computer 52 (1) (2019) 36–45.

[2] D. Wang, T. Abdelzaher, L. Kaplan, Social sensing: building reliable systems on unreliable data, Morgan Kaufmann, 2015.

[3] Z. Zhang, Q. He, J. Gao, M. Ni, A deep learning approach for detecting traffic accidents from social media data, Transportation research part C: emerging technologies 86 (2018) 580–596.

[4] T. H. Nazer, G. Xue, Y. Ji, H. Liu, Intelligent disaster response via social media analysis a survey, ACM SIGKDD Explorations Newsletter 19 (1) (2017) 46–59.

[5] Y. Mejova, I. Weber, L. Fernandez-Luque, Online health monitoring using facebook advertisement audience estimates in the united states: evaluation study, JMIR public health and surveillance 4 (1) (2018) e30.

[6] D. Wang, L. Kaplan, T. Abdelzaher, C. C. Aggarwal, On credibility estimation tradeoffs in assured social sensing, IEEE Journal on Selected Areas in Communications 31 (6) (2013) 1026–1037.

[7] D. T. Nguyen, F. Ofli, M. Imran, P. Mitra, Damage assessment from social media imagery data during disasters, in: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, 2017, pp. 569–576.

[8] X. Li, D. Caragea, C. Caragea, M. Imran, F. Ofli, Identifying disaster damage images using a domain adaptation approach.

[9] P. Kumar, F. Ofli, M. Imran, C. Castillo, Detection of disaster-affected cultural heritage sites from social media images using deep learning techniques, Journal on Computing and Cultural Heritage (JOCCH) 13 (3) (2020) 1–31.

[10] H. Mouzannar, Y. Rizk, M. Awad, Damage identification in social media posts using multimodal deep learning., in: ISCRAM, 2018.

[11] X. Li, D. Caragea, H. Zhang, M. Imran, Localizing and quantifying damage in social media images, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 194–201.

[12] D. Zhang, Y. Zhang, Q. Li, T. Plummer, D. Wang, Crowdlearn: A crowd-ai hybrid system for deep learning-based damage assessment applications, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2019, pp. 1221–1232.

[13] T. Elsken, J. H. Metzen, F. Hutter, et al., Neural architecture search: A survey., J. Mach. Learn. Res. 20 (55) (2019) 1–21.

[14] D. McDuffie, Using amazon's mechanical turk: benefits, drawbacks, and suggestions, APS Observer 32 (2) (2019).

[15] B. Zoph, V. Vasudevan, J. Shlens, Q. V. Le, Learning transferable architectures for scalable image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8697–8710.

[16] H. Liu, K. Simonyan, Y. Yang, Darts: Differentiable architecture search, in: International Conference on Learning Representations, 2018.

[17] Y. Li, R. Ji, S. Lin, B. Zhang, C. Yan, Y. Wu, F. Huang, L. Shao, Dynamic neural network decoupling, arXiv preprint arXiv:1906.01166 (2019).

[18] G. Neubig, C. Dyer, Y. Goldberg, A. Matthews, W. Ammar, A. Anastasopoulos, M. Ballesteros, D. Chiang, D. Clothiaux, T. Cohn, et al., Dynet: The dynamic neural network toolkit, arXiv preprint arXiv:1701.03980 (2017).

[19] F. Alam, F. Ofli, M. Imran, M. Aupetit, A twitter tale of three hurricanes: Harvey, irma, and maria, arXiv preprint arXiv:1805.05144 (2018).

[20] D. Y. Zhang, C. Zheng, D. Wang, D. Thain, X. Mu, G. Madey, C. Huang, Towards scalable and dynamic social sensing using a distributed computing framework, in: Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on, IEEE, 2017, pp. 966–976.

[21] D. Sahoo, Q. Pham, J. Lu, S. C. Hoi, Online deep learning: Learning deep neural networks on the fly, arXiv preprint arXiv:1711.03705 (2017).

[22] D. T. Nguyen, S. Joty, M. Imran, H. Sajjad, P. Mitra, Applications of online deep learning for crisis response using social media information, arXiv preprint arXiv:1610.01030 (2016).

[23] A. Wan, X. Dai, P. Zhang, Z. He, Y. Tian, S. Xie, B. Wu, M. Yu, T. Xu, K. Chen, et al., Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12965–12974.

[24] N. Q. V. Hung, N. T. Tam, L. N. Tran, K. Aberer, An evaluation of aggregation techniques in crowdsourcing, in: International Conference on Web Information Systems Engineering, Springer, 2013, pp. 1–15.

[25] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, Q. V. Le, Mnasnet: Platform-aware neural architecture search for mobile, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2820–2828.

[26] Y. Zhang, R. Zong, Z. Kou, L. Shang, D. Wang, A crowd-driven dynamic neural architecture searching approach to quality-aware streaming disaster damage assessment, in: 2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS), IEEE, 2021, pp. 1–6.

[27] D. Wang, L. Kaplan, H. Le, T. Abdelzaher, On truth discovery in social sensing: A maximum likelihood estimation approach, in: Proceedings of the 11th international conference on Information Processing in Sensor Networks, 2012, pp. 233–244.

[28] D. Wang, L. Kaplan, T. F. Abdelzaher, Maximum likelihood analysis of conflicting observations in social sensing, ACM Transactions on Sensor Networks (ToSN) 10 (2) (2014) 1–27.

[29] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, P. Bouvry, A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities, IEEE communications surveys & tutorials 21 (3) (2019) 2419–2465.

[30] D. Zhang, D. Wang, N. Vance, Y. Zhang, S. Mike, On scalable and robust truth discovery in big data social media sensing applications, IEEE transactions on big data 5 (2) (2018) 195–208.

[31] D. Zhang, D. Wang, H. Zheng, X. Mu, Q. Li, Y. Zhang, Large-scale point-of-interest category prediction using natural language processing models, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 1027–1032.

[32] Y. Zhang, R. Zong, Z. Kou, L. Shang, D. Wang, Collablearn: An uncertainty-aware crowd-ai collaboration system for cultural heritage damage assessment, IEEE Transactions on Computational Social Systems (2021).

[33] D. Y. Zhang, Y. Huang, Y. Zhang, D. Wang, Crowd-assisted disaster scene assessment with human-ai interactive attention, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 2717–2724.

[34] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for iot big data and streaming analytics: A survey, IEEE Communications Surveys & Tutorials 20 (4) (2018) 2923–2960.

[35] L. Zhang, X. Sun, Y. Li, Z. Zhang, A noise-sensitivity-analysis-based test prioritization technique for deep neural networks, arXiv preprint arXiv:1901.00054 (2019).

[36] D. C. Brabham, Crowdsourcing, Mit Press, 2013.

[37] D. K. Harris, M. Alipour, S. T. Acton, L. R. Messeri, A. Vaccari, L. E. Barnes, The citizen engineer: Urban infrastructure monitoring via crowdsourced data analytics, in: Structures Congress 2017, 2017, pp. 495–510.

[38] F. J. C. Dos Reis, S. Lynn, H. R. Ali, D. Eccles, A. Hanby, E. Provenzano, C. Caldas, W. J. Howat, L.-A. McDuffus, B. Liu, et al., Crowdsourcing the general public for large scale molecular pathology studies in cancer, EBioMedicine 2 (7) (2015) 681–689.

[39] D. Y. Zhang, Q. Li, H. Tong, J. Badilla, Y. Zhang, D. Wang, Crowdsourcing-based copyright infringement detection in live video streams, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 367–374.

[40] X. Wang, X. Zheng, Q. Zhang, T. Wang, D. Shen, Crowdsourcing in its: The state of the work and the networking, IEEE transactions on intelligent transportation systems 17 (6) (2016) 1596–1605.

[41] T. Mo, Y. Yu, M. Salameh, D. Niu, S. Jui, Neural architecture search for keyword spotting, arXiv preprint arXiv:2009.00165 (2020).

[42] M. Zhou, Z. Bai, T. Yi, X. Chen, W. Wei, Performance predict method based on neural architecture search, Journal of Internet Technology 21 (2) (2020) 385–392.

[43] T. Shermin, S. W. Teng, M. Murshed, G. Lu, F. Sohel, M. Paul, Enhanced transfer learning with imagenet trained classification layer, in: Pacific-Rim Symposium on Image and Video Technology, Springer, 2019, pp. 142–155.

[44] D. Wang, T. Abdelzaher, L. Kaplan, C. C. Aggarwal, Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications, in: 2013 IEEE 33rd international conference on distributed computing systems, IEEE, 2013, pp. 530–539.

[45] C. Wang, D. Chen, L. Hao, X. Liu, Y. Zeng, J. Chen, G. Zhang, Pulmonary image classification based on inception-v3 transfer learning model, IEEE Access 7 (2019) 146533–146541.

[46] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks., in: CVPR, Vol. 1, 2017, p. 3.

[47] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[48] R. Artstein, M. Poesio, Inter-coder agreement for computational linguistics, Computational Linguistics 34 (4) (2008).

[49] G. Jurman, S. Riccadonna, C. Furlanello, A comparison of mcc and cen error measures in multi-class prediction, PloS one 7 (8) (2012) e41882.

[50] D. Chicco, G. Jurman, The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation, BMC genomics 21 (1) (2020) 6.