Private and Scalable Personal Data Analytics using a Hybrid Edge-Cloud Deep Learning

Seyed Ali Osia, Ali Shahin Shamsabadi, Ali Taheri, Hamid R. Rabiee, Nicholas D. Lane, Hamed Haddadi

Abstract—We are observing an increasing presence of cyber-physical systems and their associated data around us. While the ability to collect, collate, and analyze the vast amount of rich information from smartphones, IoT devices, and urban sensors can be beneficial to the users and the industry, this process has led to a number of challenges ranging from performing efficient and meaningful analytics on the generated big data, to privacy challenges associated with the inferences made from these data due to ubiquitous nature of connected devices.

In this paper, we discuss novel edge-computing methods to improve the scalability and privacy of user-centered analytics. We present a hybrid framework where edge devices and resources centered around the user can complement the cloud for providing privacy-aware, yet accurate and efficient analytics. We present early evaluations of the proposed framework on a number of exemplar applications, and discuss the broader implications of such approaches.

Index Terms—Deep Learning, Edge Computing, Privacy.

1 Introduction

The rapid rise in the development and implementation of cyber-physical systems and the Internet of Things (IoT) devices are transforming our interaction with the physical world. Today, smart devices and ambient sensors are pervasively and continuously collecting and transferring large volumes of diverse user data for a variety of purposes including security surveillance, health monitoring, and urban planning. Today, majority of IoT devices are constantly *online* by default and rely on machine learning applications over the cloud in order to gain insights from the collected data. Sophisticated corporate cloud computing services provide on-demand high performance, efficient computational power and considerable cost reduction.

Despite all its benefits, cloud computing comes with certain challenges. Mobile and broadband bandwidth and efficiency will be a major bottleneck as the *smart home* and *smart car* of the next decade will be uploading vast volume of data from hundreds of sensors to cloud processors. The cloud-based models will also impose major energy constraints on the edge devices. Privacy issues are another important threat posed by cloud-based systems; users risk their potential ownership of sensitive data by sharing it and allowing the service providers to harvest, analyze, or monetize their data. for example, majority of cloud-based mobile applications are free, relying on information harvesting from their users' personal data for targeted advertising. This practice has a number of privacy implications and resource impacts for the users [12]. Preserving individuals' privacy, versus detailed data analytics, face a dichotomy in this space. Cloud-based machine learning algorithms can provide beneficial or interesting services (*e.g.* , health or image-based search mobile applications), albeit, their reliance on excessive data collection from the users can have consequences which are unknown to the user (*e.g.* , face recognition for targeted social advertising).

Recently, *Edge Computing* has been proposed to tackle these issues by locating the processing power in edge nodes, or near to the end user in the similar context of *Fog Computing* at the network edge. In this way, delay-sensitive data can be analyzed on the edge nodes and cloud services can be leveraged for more delay-tolerant tasks. Yet, an analytics service or an app provider might not be keen on sharing their valuable and highly-tuned data processing models. Hence, it is not always possible to assume feasibility of local processing (*e.g.* , a deployed deep learning model on an edge device such as a smartphone or a computer) is a viable solution even if the task duration, memory and processing requirements are not important for the user or tasks can be performed when the user is not actively using their devices (*e.g.* while the device is being charged overnight).

One may suggest that fully cryptographic-based algorithms are the ideal solution to these concern; however the complexity of encryption methods can be high for many IoT applications, specially the ones relying on local machine learning models, or modules that need to be continuously available or online (e.g., multimedia applications or sensors in a self-driving vehicle). Deep learning models are highly non-linear, complex functions; these are difficult to estimate with polynomial functions which are an essential component of cryptographic-based algorithms such as homomorphic encryption [6].

On one hand, complete data offloading to cloud services can have immediate or future scalability and privacy risks; on the other hand, techniques relying on performing

- *Seyed Ali Osia, Ali Shahin Shamsabadi, Ali Taheri, and Hamid R. Rabiee are with the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.*
- *Nicholas D. Lane is withNokia Bell Labs and UCL, UK.*
- *Hamed Haddadi is with the Dyson School of Design Engineering at Imperial College London, UK.*

Manuscript received September 15, 2017; revised ...

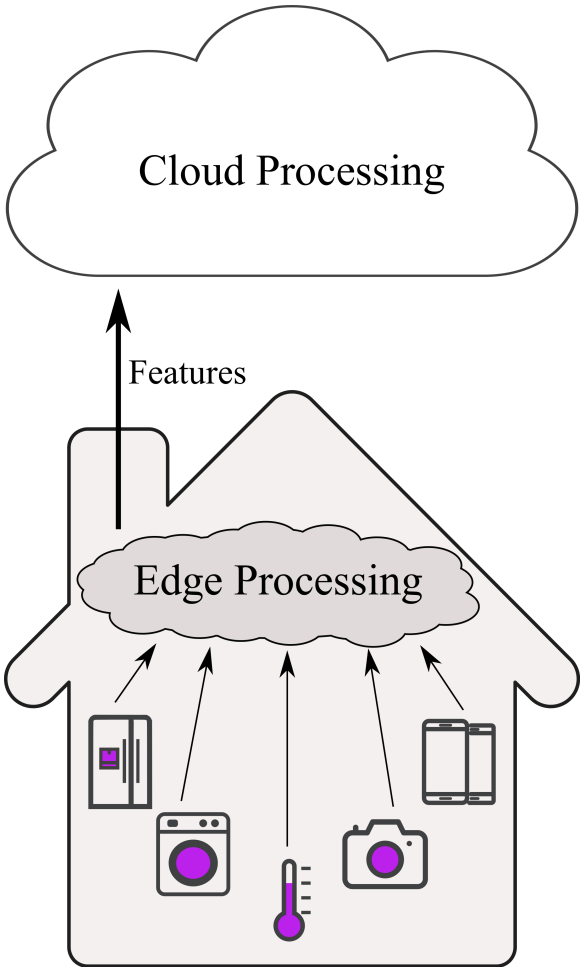

Fig. 1. Hybrid edge-to-cloud framework for privacy-preserving machine learning. User's data is collected and processed locally on private edge nodes in order to preserve sensitive information. The representation of data which is independent of sensitive information is sent to cloud data center for applying complex inferences.

complete analytics at the user end, or encryption-based methods, also come with their own resource constraints (*e.g.* storage and bandwidth constraints, energy limitations, or computational costs) and user experience penalties.

In this paper, we design a hybrid edge-to-cloud architecture, in which data processing is accomplished in collaboration between private edge data processing units and cloud services. In this way, we can augment the end user to benefit from the cloud processing efficiency while addressing the privacy concerns by leveraging edge preprocessing. A schematic view of this framework is shown in Fig. 1. We focus on achieving a compromise between resource-hungry local analytics on a private edge node, versus data hungry and privacy-invasive cloud-based services. The least necessary amount of processing will take place in the edge node, which assures privacy preservation, while the rest of processing occurs in the cloud. Our main objective is to separate the feature extraction and the inference phase; the former takes place locally, while the latter takes place on the cloud. With this approach, while reducing the data transmission rates to the cloud, sensitive information can be removed from the data during the feature extraction phase on the edge node. The extracted features are then transferred to the cloud server for post-processing and finally the user receives the results from the cloud.

2 Real World Applications

Advances in computer vision, machine learning and cloud computing techniques have provided new opportunities in a large number of multimedia IoT services [1]. In this paper, we explore the network bandwidth and privacy challenges faced by these cloud-based multimedia IoT applications in the following domains:

- Image Processing: The increasing quality of smartphone cameras and sensors, in addition to the rise in popularity of image-centric social media, have all led to a variety of image analytics applications, such as scene tagging, image classification, face recognition, facial attribute prediction, age estimation, gender classification, and emotion recognition.
- Video Processing: Excessive presence of CCTV camera shows the importance of video recording, indexing and processing. Many homes and outdoor environments are equipped by video surveillance systems to capture visual information for different purposes. Smart cameras are used in care homes to provide health care services and elderly monitoring. Cameras in autonomous vehicles and monitoring the vehicles in highways or parking lots is another application of video processing in public places.
- Speech Processing: Speech is increasingly becoming used in human-device interaction in the IoT domain. Many smart televisions, phones, watches, ovens and lights have voice command features. Increasingly, devices like Google Home and Amazon Echo are now entering the households as intelligent home assistants. In the next few years, speech recognition systems will become an integral part of our life.

All the above applications need sophisticated processing of large volumes of data, usually achieved by machine learning algorithms. As an example, consider a classification problem such as face recognition. The classification model should be trained with a large training dataset consisting of face photos, each labeled with the person's identity. After training, the model is able to label a face photo with its identity. In general, the machine learning problems are categorized into *supervised*, *unsupervised*, and *semi-supervised*. In supervised problems, true labels for training data is available, and the goal is to predict the label of a test data, similar to the aforementioned example. In this paper, we focus on supervised applications, specially classification. Interested readers can refer to [2] to obtain more knowledge about machine learning. When true labels are not accessible, the problem is referred to as unsupervised learning, *e.g.* clustering. When a small number of labeled data and an abundance of unlabeled data is available, semi-supervised methods utilize the unlabeled data to enhance the result of supervised classification based on the few labeled data.

In all of the aforementioned applications, an operator might be concerned about transferring the large volume of IoT data produced at the edge on the broadband or mobile

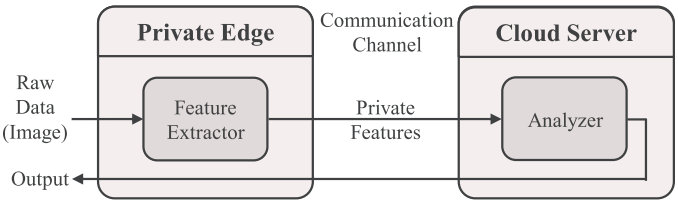

Fig. 2. Modules of the proposed framework. The Analyzer in the cloud server has access to reduced set of private features of the data, which are provided by the Feature Extractor placed in the edge node.

network, and clients are concerned about disclosure of their sensitive information. In many applications, significant part of clients' data does not need to be recognized by a service provider [3]. In surveillance or analytic applications, the individuals' identity is the most sensitive information that has been collected. For example, an individual walking by a number plate recognition camera in a car park should not be identifiable, while classification or optical character recognition techniques are being applied to the plate. In other words, the individuals might want to be protected against undesired face recognition models. Similarly, an individual using an IoT device voice prompt might wish to be unidentifiable through their voice sessions. Privacy concerns also rise in health analytics, when the application users might not wish to reveal their private information. These privacy concerns show the value and importance of a general framework that is capable of addressing the privacy issues.

3 Framework Abstraction

Let us assume we wish to execute a primary task (*e.g.* , speech recognition or image analysis) via cloud services, with constraints due to limited local processing capabilities or conflicting commercial reasons. On the other hand, we wish to preserve sensitive user information (*e.g.* , the identity of the speaker which could be disclosed through his voice, or an individual's pictures in an urban CCTV image). Hence, the data shared with the cloud service should possess two important properties: (i) inferring the primary task is possible; and (ii) deducing sensitive information is not possible.

Sharing data on the cloud provides the probability of further inferences made on sensitive information. Edge-based pre-processing of the raw data can prevent revealing undesired features of the data, however such a task needs to have minimal burden due to various limitations in client side. In order to achieve this, we propose a general hybrid architecture which contains two main modules: a *Feature Extractor*, and an *Analyzer*, where the former is constructed in a private edge node (like a personal computer or home set-top box), and the later is held in the cloud. These modules and their interaction are shown in Fig. 2. The data from client devices are collected one the private edge node and sent to the Feature Extractor, which gets the input data, applies a function on it, and outputs a set of new *intermediate features*, which would be then transferred to the cloud for performing the primary tasks. The Analyzer receives the intermediate features, infers the primary information, and if needed, returns back the result to the client side.

In this framework, designing a good Feature Extractor module is the of critical importance. The intermediate features need to keep the necessary information about the primary task, and on the other hand they should protect the sensitive information. As the Feature Extractor operates locally, it should not be a complex routine; hence designing this module is a challenging and important task.

As a use case, we consider an image tagging cloud service, in which the sensitive information is the identity of the individuals in an image from a live video stream. In this case, a simple Feature Extractor can just detect faces and replace them with shaded regions. The Analyzer receives this censored image and performs the image tagging procedure (*e.g.* label the image with objects, places, pets, etc.). Another common example is speech recognition, in which we might be concerned about being identified through our voice tone. One simple solutions is to simply pitch frequency of the voice in the Feature Extractor to achieve anonymity. In these two cases, designing the Feature Extractor is simple and will not affect the Analyzer's result; however, this is not always the case.

In the above examples, we demonstrated simple approaches to remove a part of the data which contains sensitive information, and then considered the remaining part as intermediate features. However, this is not applicable when the removing part contains important information about the primary task. For example, facial attributes like emotion or gender get disposed at the same time as removing sensitive information (the identity) by blocking a face region. Hence we can not use this method when our primary task, is for example, facial attribute prediction.

When the primary and sensitive information are interlocked, we encounter a complex situation. In this case, unlike the previously discussed examples, we should also consider the primary task in designing the Feature Extractor module for sensitive information removal. In our framework we present a method based on deep learning, which considers both primary task and sensitive information in the design procedure. Assuming the service provider knows about the type of client sensitive information (*e.g.* identity), the following scenario occurs: The service provider hands over a Feature Extractor module to the client, which is guaranteed to care about the primary task and the sensitive information, simultaneously. While the service provider does not have to share the Analyzer, it must define a verification method for the privacy preservation. This process defines a privacy standard that the service providers should adopt.

4 Deep Learning Applications

Deep neural networks have become popular in machine learning, specially in multimedia applications [7]. They provide highly accurate classifiers that extract high level information from raw data. Deep networks consist of different layers that follow each other. Each layer is a simple function of its previous layer, representing a more sophisticated concept than its previous layers. The initial layer is the raw input data and the final layer gives the result of inference. All these layers together, form a complex function which

is applied to the input data and results in a perceptual inference. The intermediate functions are learned during the training phase, via applying optimization methods on the training data. When the model is trained, it is ready to perform inference on any input data.

Deep Convolutional Neural Networks (CNNs) and deep Recurrent Neural Networks (RNNs) are the two most famous structures used for multimedia applications. The former is suitable for image and video processing, and the latter is designed mainly for sequential data processing (*e.g.* streaming speech and video). In this paper, we focus on CNNs as the most popular structure for image and video processing. Suppose inference about a primary task is done with a pre-trained deep network (*i.e.*, a ready to use network with many layers). We address how to embed this trained model in the proposed edge-to-cloud framework as follows.

4.1 Layer Separation

In deep models, the higher layers become more and more specific to the primary task, while losing other irrelevant information that contains the sensitive information that we are concerned with. Based on this observation, we propose a layer separation mechanism for a pre-trained deep network:

- First, choose an intermediate layer as the separation point.
- Then, store the layers before the intermediate layer on the edge node as the Feature Extractor.
- Finally, store the layers after the intermediate layer on the cloud server as the Analyzer.

There is a trade-off on selecting the intermediate layer; Choosing it from higher layers results in higher privacy for sensitive information. However, this selection also increases the computational costs on the client side. We refer to this simple separation of layers between the edge and cloud as the *simple embedding* as shown in Figure 3.

4.2 Siamese Embedding

In order to increase the privacy when revealing the intermediate feature to the server, we can fine-tune the existing deep model for the primary task with a particular method. Fine-tuning is a common task in training deep models. We start from a pre-trained deep model and continue its training to achieve a desired goal. As a result, we obtain an updated trained deep model, which can be used in the layer separation mechanism.

The main novelty of the proposed method relies on fine-tuning the model of primary task by utilizing the Siamese architecture [4], based on the chosen intermediate layer. Siamese architecture is a common way of training learning models, with a popular usage in face verification applications, where we are trying to decide whether two images belongs to the same person or not. As a result, we construct a feature space, where similar points group together. The main idea behind the Siamese network is forcing the representations of *similar* points (*e.g.* different face images from the same person) to become near to each other, and the representations of *dissimilar* points (*e.g.* face images from different persons) to become far from each other.

To achieve this goal, our training dataset should consist of pairs of points, which can be similar or dissimilar. For a pair of points, one function is applied to both of them and the distance of two outputs is computed. Optimization is done based on a contrastive loss function. For this loss function, the distance is maximized for two dissimilar points and minimized for two similar points. This approach makes the Feature Extractor more private and preserves the users' privacy against inference attacks on the cloud. We refer to this embedding as the *Siamese embedding*.

4.3 The Siamese Architecture Privacy

How can we relate the Siamese architecture to privacy? Without loss of generality, we answer this question with a clarifying example, which shows how we can indirectly use the Siamese contrastive to obtain more privacy.

Suppose our primary task is gender recognition through face portraits, accomplished by a pre-trained deep model. The sensitive information is our identity which should not be disclosed by using the intermediate data (*e.g.* by a face recognition system). In this scenario, the only thing we care about is gender of the face portrait and not its identity. We can model this fact by defining a new similarity criterion and then fine-tune our model with a contrastive loss function. Considering all identities with the same gender as similar, not only makes the gender recognition model more robust, but also eliminates more identity information from the intermediate features. After fine-tuning with this method, men representations are very close to each other while being far from the women representations, which are also close to each other.

Fine-tuning structure for privacy preservation is shown in Figure 4. We can apply this idea to any application by appropriately defining the similarity criterion. Experiments show that using the Siamese embedding improves the privacy preservation while maintaining the accuracy of the primary task.

4.4 Dimensionality Reduction

An important issue about all cloud-based services is their communication cost which is usually too high. We are going to address this concern by reducing the dimensionality of the intermediate features.

Dimensionality reduction has a long range of applications in statistics and machine learning; from visualization to feature extraction. The dimensionality of data can be reduced by linear or nonlinear transformations of a high dimensional space to a lower one. One of the most popular dimensionality reduction methods is the Principal Component Analysis (PCA). PCA uses an orthogonal linear transformation for keeping the highest variance along the desired dimensions of the data. The reduction and reconstruction procedure can be achieved by matrix multiplication.

The Siamese fine-tuning makes feature space much more robust in a way that applying PCA on the fine-tuned space does not significantly decrease the accuracy of the primary task. Using dimensionality reduction on the intermediate feature space brings us two advantages without a significant reduction in primary task accuracy: (i) it highly reduces the edge-to-cloud communication cost, and (ii) it highly

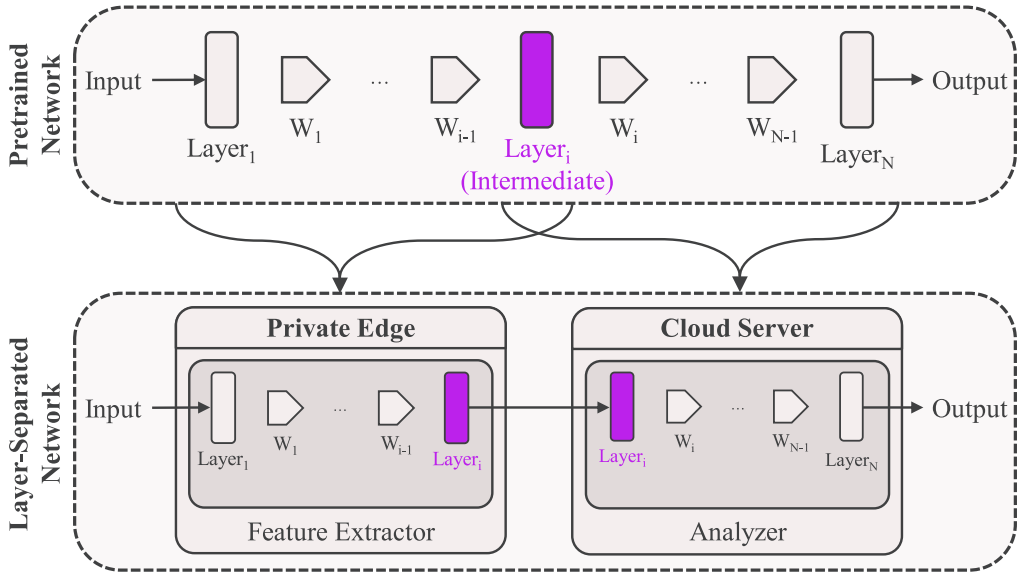

Fig. 3. Layer Separation mechanism. Primary layers of the deep network correspond to Feature Extractor and the rest of the model is considered as Analyzer.

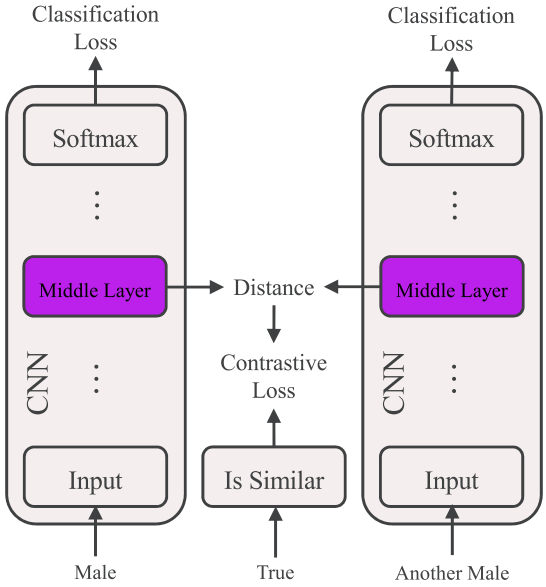

Fig. 4. Siamese fine-tuning of primary task deep model. Intermediate features of two male face images are extracted via two identical CNNs. They should be close to each other because they are considered as similar.

increases the privacy based on the nature of reduction-reconstruction procedure.

The process of applying PCA on the intermediate feature is as follows. The service provider adds the PCA projection and reconstruction at the end of the Feature Extractor and start of the Analyzer, respectively. Hence the extracted intermediate feature would be a low dimensional vector which can be easily transferred to the cloud with low communication cost. By using these two methods, we introduce *Advanced embedding*, in which Siamese fine-tuning is added as a pre-process and PCA projection is applied on the intermediate feature.

5 Preliminary Evaluations

We have performed extensive experiments on face images, corresponding to the aforementioned gender classification problem as the primary task, and identity of the participated person as the sensitive information to be preserved. For each of the suggested embeddings, we evaluated the amount of information that the intermediate feature has about gender and identity. We used an intuitive visualization technique, which demonstrates to what extend it is possible to reconstruct the original image from the intermediate data representation. We have also employed more rigorous analysis of our proposed approach in [8], where we proposed a privacy measure in order to formally quantifying the ability of this framework to preserve sensitive information.

In order to compare different deep embeddings, we used the gender classification model proposed by Rothe *et al.* [9]. This model is a 16-layer CNN with the popular VGG-16 structure [11]. They collected a large face dataset, containing age and gender attributes from IMDB and Wikipedia. Their model achieved 93% accuracy on the Wikipedia images. To provide a fair comparison, we have also performed our experiments on this dataset.

We chose the fifth convolutional layer as our intermediate feature. Simple embedding needs nothing more than layer separation. Siamese embedding is done by fine-tuning the pre-trained model and then doing the layer separation. Advanced embedding has also the same procedure with an additional process for applying PCA. We reduced the dimensions of the intermediate feature to eight. We analyzed the tradeoff between the accuracy of the gender classification (primary task) and the privacy of identity (sensitive information). Surprisingly, all these embeddings reached almost the same 93% accuracy of gender classification on average. Therefore, they all had similar performances in satisfying the primary task. Hence, the only critical issue for comparison is their ability to maintain more privacy through their identity preservation capability.

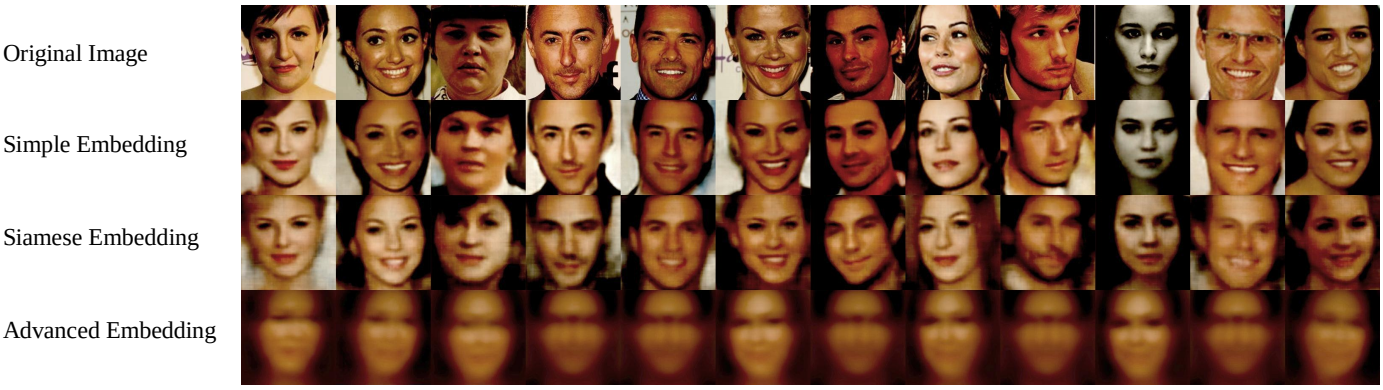

Fig. 5. Comparison of different deep embeddings for privacy preservation by using visualization. All methods had similar gender classification accuracy of 93%. The first row shows the original images and the others show the reconstructed ones from intermediate representations. In all reconstructed images, the gender of the individuals is recognised to be the same as the originals. In addition, From simple to advanced embedding, the identity of the individuals is increasingly removed, illustrating that the *advanced embedding* has the best privacy-preservation performance.

We compared the ability of these methods in privacy preservation by using a visualization technique. Visualization tries to answer a key question: Having just the intermediate layer of a deep network, what is the best recognition possibility for the original input image? The authors in [5] have answered this question by training a decoder by using the intermediate layer as its input and the generating image as its desired output. We used their method and compared the results for different deep embeddings (although it can not be considered as a rigorous proof for superior performance, it is highly intuitive).

The restored original images from intermediate features are illustrated in Fig. 5 for different methods. It can be observed that the genders of all images in the simple and Siamese embeddings remain the same as the original ones. This is also the case for the advanced embedding because of the accuracy of gender classification, although it is harder to distinguish it from the reconstructed images. The original images are almost restored in the simple embedding. Therefore, just separating layers of a deep network can not assure acceptable privacy preservation performance. Siamese embedding performs better than the simple embedding by distorting the identity due to intrinsic characteristics of the face (*e.g.* the skeleton). Finally, the Advanced Embedding provides the best results, because the decoder was not trainable and nothing can be deduced from intermediate images, including the person's identity. As an advantage of this method, the communication cost is really negligible compared to other cases, because we needed to upload only 8 real numbers to the cloud. Further detailed analysis are presented in [8].

6 Conclusions

In this paper, we presented a new hybrid edge-to-cloud framework for efficient, privacy preserving analytics on multimedia or IoT applications. Our framework consists of a Feature Extractor and an Analyzer module, where the former is placed on the edge device and the latter on the cloud. We described how to use this framework in various IoT multimedia applications and studied deep learning as a popular special case. We embedded deep neural networks, specially convolutional neural networks in this framework to benefit from their accuracy and layered architecture. In order to protect the data privacy against unauthorized tasks, we used the Siamese architecture, creating a feature specific to the desired task. This is in contrast to today's ordinary deep networks in which the created features are generic and can be used for different tasks. Removing the undesired information from the extracted feature results in achieving privacy preservation for the user.

Our framework is currently designed for pre-trained machine learning inferences. In ongoing work we aim to extend our method by designing a framework for *Learning-as-a-Service* [10], in which users could share their data, in a privacy-preserving manner, for training a new learning model in the cloud server. Another potential extension to our framework will be providing support for other kinds of neural networks such as recurrent neural network which is useful for temporal and sequential data processing.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. A survey on wireless multimedia sensor networks. *Computer networks*, 51(4):921–960, 2007.

[2] C. M. Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

[3] A. Chaudhry, J. Crowcroft, H. Howard, A. Madhavapeddy, R. Mortier, H. Haddadi, and D. McAuley. Personal data: Thinking inside the box. In *Proceedings of The Fifth Decennial Aarhus Conference on Critical Alternatives*, AA '15, pages 29–32. Aarhus University Press, 2015.

[4] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[5] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4829–4837, 2016.

[6] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210, 2016.

[7] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[8] S. A. Osia, A. S. Shamsabadi, A. Taheri, H. R. Rabiee, N. Lane, and H. Haddadi. A hybrid deep learning architecture for privacy-preserving mobile analytics. *arXiv preprint arXiv:1703.02952*, 2017.

[9] R. Rothe, R. Timofte, and L. Van Gool. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–15, 2015.

[10] S. Servia-Rodriguez, L. Wang, J. R. Zhao, R. Mortier, and H. Haddadi. Personal model training under privacy constraints. *arXiv preprint arXiv:1703.00380*, 2017.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[12] N. Vallina-Rodriguez, J. Shah, A. Finamore, Y. Grunenberger, K. Papagiannaki, H. Haddadi, and J. Crowcroft. Breaking for commercials: Characterizing mobile advertising. In *Proceedings of the 2012 Internet Measurement Conference*, IMC '12, pages 343–356, New York, NY, USA, 2012. ACM.