Imperial College London
Hamlyn Centre for Robotic Surgery
Department of Computing

Decentralised Federated Machine Learning at the Edge on IoT Devices for Healthcare

James Calo

Supervised by: Dr. Benny Lo

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy at Imperial College London, December 2024

Declaration of Originality

I declare that the research contained within this thesis is my own and was undertaken as a PhD student at Imperial College London. All content that was not produced by me has been appropriately referenced and this thesis has not been submitted for another degree.

Copyright Declaration

Abstract

Deep learning has revolutionised healthcare, transforming the analysis of medical data and achieving performance on par with medical experts in a wide range of imaging tasks. This paradigm shift has established artificial neural networks as pivotal tools in healthcare applications, including disease diagnosis, treatment planning, and surgical interventions. As such, machine learning has become a cornerstone of modern medical innovation, driving progress toward improved clinical outcomes and personalized care.

However, training deep neural networks directly on the Internet of Things (IoT) devices at the edge poses significant challenges due to their constrained computational resources. Consequently, the vast potential of IoT devices in healthcare remains underutilized. These devices, widely available in medical environments, offer a unique opportunity to decentralise artificial intelligence (AI) by shifting computation from centralised cloud systems to the edge. Exploiting mist architectures already deployed in hospitals, this approach has the potential to revolutionise healthcare by enabling real-time processing, reducing latency, and enhancing data privacy and security.

Despite this promise, the adoption of machine learning in healthcare is hindered by critical challenges, including data privacy concerns, interoperability barriers, and the need for equitable access to technology. Innovative solutions are required to address these obstacles and fully realise the transformative potential of AI-driven healthcare.

This thesis addresses these challenges by designing a novel, decentralised federated learning framework tailored for healthcare applications. This framework facilitates privacy-preserving inter-hospital collaboration by leveraging existing mist architectures and the ubiquity of IoT devices. A key innovation of this research is the development of a bespoke blockchain consensus mechanism integrated with masked autoencoders (MAEs), providing robust privacy protection and enabling advanced federated learning capabilities. This work represents a significant step toward unlocking the full potential of decentralised machine learning in healthcare, paving the way for secure, collaborative, and scalable AI solutions in clinical practice.

Acknowledgements

I wish to express my deepest gratitude to my supervisor and mentor, Dr. Benny Ping Lai Lo, for his unwavering support, insightful guidance, and generous dedication of time throughout this journey. This PhD would not have been possible without his invaluable expertise and encouragement.

I extend my heartfelt thanks to my colleagues and friends, Dr. Frank Lo, Dr. Xiao Gu, Zeyu Wang, and Junhong Chen, whose support, advice, and warm camaraderie have made my PhD experience not only intellectually rewarding but also immensely enjoyable.

I am profoundly grateful to my friends and family, especially my mother and brother, Alex. Their unwavering belief in me has been a constant source of motivation and strength. I will always cherish the pep talks with Alex that lifted my spirits during challenging moments. Lastly, a special mention goes to my cat, Piccolo, whose silent companionship and "discussions" about my research have been a unique source of comfort and inspiration.

Dedication

In loving memory of my two amazing grandmothers who supported me throughout my academic endeavours.

Contents

List of Tables

List of Figures

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
CT	Computed Tomography
EDI	Encoder-Decoder Interface
FL	Federated Learning
GAN	Generative Adversarial Network
iid	Independent and Identically Distributed
IoT	Internet of Things
IoMT	Internet of Medical Things
kNN	k-nearest neighbours
LotE	Learning on the Edge
MAE	Masked Autoencoder
MAE_ℓ	Mean Absolute Error (loss function)
ML	Machine Learning
MRI	Magnetic Resonance Imaging
MSE_ℓ	Mean Squared Error (loss function)
NN	Neural Network
P2P	Peer-to-peer
PoW	Proof of Work
PoS	Proof of Stake
PoR	Proof of Reasoning
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
roi	Region of Interest
SGD	Stochastic Gradient Descent
ViT	Vision Transformers

Chapter 1

Introduction

1.1 Motivation and Objectives

Deep learning has emerged as the dominant paradigm for artificial neural network development in recent years [5, 6] becoming a transformative force in healthcare, revolutionising how medical data is analysed and performing on par with medical experts in a multitude of medical imaging tasks [7] cementing their value in diverse healthcare applications, from disease diagnosis and treatment planning to surgical interventions.

Unfortunately, training deep networks directly on IoT devices at the edge remains impractical due to their substantial computational requirements [8, 9]. Consequentially, the vast and largely untapped computational potential of IoT devices, particularly in healthcare, is underutilised.

Training on the IoT device itself improves privacy by enabling models to train on data without requiring persistent storage. Furthermore, with training offloaded onto IoT devices energy consumption can be reduced as high computational powered devices would no longer be required and would potentially eliminate the need for cloud computing and therefore connection to a network.

These devices, ubiquitous in medical settings, represent an opportunity to decentralise artificial intelligence by shifting computation from centralised cloud systems to the edge, leveraging

existing mist architectures already deployed in hospitals. This shift has the potential to revolutionise healthcare enabling real-time processing, reducing latency, and enhancing data security.

Nevertheless, this transition is fraught with challenges. Even with access to powerful computational resources, deep learning networks are prone to overfitting on large datasets comprising vast volume of data [10]. Furthermore, successful training hinges on the availability of annotated data [11], much of which is inaccessible in healthcare due to stringent privacy regulations and data imbalances. These limitations underscore one of the most pressing challenges in machine learning: how to enhance the utility of datasets while preserving their privacy. While there exists many publicly available medical datasets online there are strict laws and guidelines that must be followed as well as ethics comities that must be convened to determine the appropriate protocol for obtaining and sharing medical data. Furthermore, data must be sanitised to remove all personally identifiable information, as such information could be used against the patient; for example in countries without free healthcare, a medical insurer could use this information to deny a policy. Additionally, in the case of rare or hard to capture medical data, such as anastomotic leaks as discussed in Chapter 3, one source of data is unlikely to be enough. Inter-hospital collaboration would be required in order to build a useful dataset which would then require all participants to agree and trust each other, an unlikely scenario even among hospitals owned by the same institution. Addressing this issue is critical for realising the potential of IoT-based federated learning in healthcare and achieving robust, decentralised AI systems at the edge.

Federated learning is uniquely suited to this task, as it can operate on non-iid (non-independent and identically distributed) data [12] and is an ideal match for the IoT environment, since it is naturally distributed with each participant training their own network and receiving a representative network after aggregation. This allows for each network to be trained on a smaller subset of the data requiring less computation complexity than training a network on a complete dataset. As a result, even if an individual hospital only has a small dataset that cannot effectively train a network, collaborating with other hospitals, in a privacy-preserving manner, produces a network akin to one trained directly on all the data.

Currently, federated learning uses either a centralised server for network aggregation or a decentralised mechanism, predominately via blockchain [13]. Centralised servers are less complex but require trust; this may be suitable when all networks belong to the server's owner but causes issues with privacy concerns when the participating network's owners have differing goals, for example, a private hospital, a public hospital and a medical research team. Therefore, decentralised (blockchain) federated learning has the most potential; however, while applications such as cryptocurrency are enhanced by the transparency inherent in blockchain, this decreases the privacy for federated learning. Not only are all networks publicly visible, and therefore can be subjected to model inversion attacks. The network suppliers may be malicious. Using current approaches, it is difficult to detect malicious networks and determine whether incorrect information has been supplied alongside the network. For example, federated averaging [14] requires the participants to include the number of data samples the network has seen which is unverifiable.

1.2 Overview of Thesis

This thesis investigates how Machine Learning (ML), particularly artificial neural networks, can be effectively deployed in healthcare applications where data is limited, imbalanced, or fragmented. Using anastomotic leak detection as a use case, the thesis begins by addressing the key challenges in developing automated diagnostic systems for such leaks. These challenges include variability in clinical definitions, inconsistencies in diagnostic procedures, and the delayed manifestation of leaks, which may only become apparent days after surgery.

To tackle these issues, this research explores the feasibility of generating 3D models from images of the anastomosis. This approach not only increases the training data available for other healthcare applications but also introduces a novel method for visualising the anastomotic joint. By enabling exploration without relying on advanced imaging modalities like Computed Tomography (CT) or Magnetic Resonance Imaging (MRI), the proposed method presents a cost-effective and accessible alternative for healthcare settings.

However, a key limitation identified throughout this process was the scarcity of labelled medical data, compounded by privacy concerns that restrict inter-hospital data sharing. To overcome these barriers, this thesis investigates federated learning, a privacy-preserving framework that allows multiple institutions to collaborate on training ML models without sharing sensitive data. To further strengthen the framework, blockchain technology is integrated, creating a decentralised system that requires no trust between participants. This combination ensures data security while enabling effective ML model development across institutions.

A significant contribution of this thesis is the demonstration of federated learning on Internet of Things (IoT) devices, showing the potential for edge-based neural network training in medical environments. This work establishes the feasibility of using IoT devices, already prevalent in many hospitals, to perform distributed training, thereby decentralising computational workloads and enhancing system scalability.

To fully integrate these advancements into a hospital's operational framework, a novel blockchain consensus mechanism tailored for federated learning was developed. This mechanism leverages existing hospital infrastructure to maximise efficiency, enabling fully distributed network training at the edge. By uniting federated learning, IoT edge computing, and blockchain technology, this research presents a comprehensive, privacy-preserving framework designed to address critical challenges in modern healthcare, laying the groundwork for broader adoption of AI-driven solutions in clinical practice.

1.3 Thesis Structure

This thesis is organized into six chapters, each building upon the previous to explore how Federated Machine Learning (FL) at the edge on IoT devices can address critical challenges in healthcare, particularly in privacy-preserving collaboration and efficient neural network training.

- Chapter 1: The introduction sets the stage by exploring the current trends in artificial

intelligence, with a focus on deep learning. It discusses the strengths and limitations of these approaches in healthcare and IoT environments while introducing the foundational technologies that underpin this research.

- Chapter 2: This chapter provides a comprehensive literature review, summarising the core technologies examined in subsequent chapters. It highlights related research addressing similar objectives, critically analysing their methodologies, and identifying the influences and distinctions that shaped this work.

- Chapter 3: Focused on the vital task of anastomotic leak detection, this chapter explores what makes detecting these leaks so critical and the limitations of relying on advanced imaging modalities such as CT and MRI. It investigates the potential of image-to-model generation for enhancing diagnostics while identifying the challenges that must be overcome to develop a fully realised system.

- Chapter 4: This chapter outlines the steps taken to address these challenges, introducing the use of federated learning and blockchain technology to train neural networks at the edge on IoT devices with limited, unbalanced data. It demonstrates the feasibility of combining these technologies to preserve privacy while enabling collaborative learning.

- Chapter 5: Building on the groundwork laid in Chapter 4, this chapter presents a novel integration of federated learning and blockchain. It introduces a bespoke consensus mechanism designed specifically for federated learning, offering a paradigm for hospitals to collaborate securely and distribute network training across existing IoT mist networks.

- Chapter 6: The final chapter summarises the key findings and outcomes of the research. It reflects on the contributions made by the thesis, discusses the implications for healthcare AI, and identifies opportunities for future research in federated learning at the edge.

This structured approach ensures a logical progression from foundational concepts to advanced solutions, culminating in a framework designed to enable secure, efficient, and collaborative AI applications in healthcare.

1.4 Contributions

This thesis makes significant contributions to the fields of machine learning, healthcare, and IoT by addressing the challenges of decentralised federated learning at the edge. The key contributions are as follows:

- Motivation and Requirements for Autonomous Healthcare Systems: a comprehensive analysis of the motivations, challenges, and requirements for developing an autonomous system for detecting anastomotic leaks, a critical complication in surgical procedures.
- Model-to-Image Generation Framework: a novel model-to-image generator for creating alternative views of an anastomotic ring, offering a viable alternative to complex imaging modalities like CT and MRI.
- Decentralised Federated Learning Framework: a robust federated learning framework for learning on the edge (LotE), demonstrating its ability to outperform traditional centralised training methods.
- Practical IoT Federated Learning System: a federated learning system on an IoT device, providing physical results that showcase the feasibility of LotE while highlighting its associated challenges in real-world healthcare environments.
- Novel Blockchain Consensus Mechanism: a new consensus mechanism, Proof of Reasoning (PoR), tailored specifically for decentralised federated learning on the blockchain. This mechanism ensures secure, privacy-preserving collaboration across multiple participants.
- Integration of Masked Autoencoders with Federated Learning: Combining masked autoencoders (MAEs) with the PoR consensus mechanism, enabling advanced federated aggregation methods that enhance privacy, data utility, and overall performance.

These contributions collectively advance the state of the art in applying machine learning at the edge in IoT-enabled healthcare settings, addressing critical challenges, such as privacy preservation, data scarcity, and computational efficiency.

1.5 Publications

The research within this thesis has resulted in publications in peer reviewed journals [**J**] and conferences [**C**]. The following publications are directly related to the research within this thesis:

1. **Calo James**, and Lo Benny. "Federated Blockchain Learning at the Edge" [**J**] In Information 14, no. 6: 318. 2023. doi: 10.3390/info14060318

2. **Calo James**, and Lo Benny. "IoT Federated Blockchain Learning at the Edge" [**C**] In Proceedings of IEEE EMBC 2023, 24-28 July 2023, Sydney Australia.
   doi: 10.1109/embc40787.2023.10339946

3. **Calo James**, and Lo Benny. "Proof of Reasoning for Privacy Enhanced Federated Blockchain Learning at the Edge" [**J**] In IEEE Internet of Things Journal (IEEE IoTJ) (Proposed, not yet accpeted)

Additionally, the following publications are not covered in this thesis but have been contributed to as a co-author:

1. Wang Zeyu, Lo Frank P.W, Huang Yunran, Chen Junhong, **Calo James**, Chen Wei, Lo Benny. "Tactile perception: A biomimetic whisker based method for clinical gastrointestinal diseases screening" [**J**] In npj Robotics 1, 3 (2023). doi: 10.1038/s44182-023-00003-8

2. Zhang Ruiyang, Chen Junhong, Wang Zeyu, Yang Ziqi, Ren Yunxiao, Shi Peilun, **Calo James**, Lam Kyle, Purkayastha Sanjay, Lo Benny Ping Lai "A Step Towards Conditional Autonomy - Robotic Appendectomy" [**J**] In The IEEE Robotics and Automation Letters, vol. 8, no. 5, pp. 2429-2436, May 2023, doi:10.1109/LRA.2023.3254859

3. Wang Zeyu, Lo Frank P.W, Chen Junhong, **Calo James**, Lo Benny, A Thompson and E Yeatman "An AI-Driven Bionic Whisker System Assisting for Clinical Gastrointestinal Disease Screening" [**C**] In Proceeding of 2024 International Joint Conference on Neural Networks (IJCNN), Yokohama, Japan 2024

Chapter 2

Background Theory

2.1 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs), introduced in 2014 by Goodfellow et al. [15], represent a transformative innovation in machine learning, particularly for image generation and domain adaptation tasks. GANs consist of two neural networks, a generator and a discriminator, that compete in a zero-sum game, enabling the generator to learn the underlying latent representation of data and produce new, realistic instances. This capability has been especially impactful in augmenting datasets by generating synthetic, yet convincing, data that can supplement model training and improve performance [16].

Initially designed for the generation of realistic images, GANs have evolved to support a wide array of tasks, including mapping between disparate data domains. Their ability to perform such mappings has opened the door to numerous applications in healthcare, such as modality-to-modality transformations. For instance, GANs can facilitate the conversion of ultrasound images into X-rays, a task that holds significant value given the challenges associated with X-rays, such as higher cost, radiation exposure, and greater technical requirements. Similarly, transforming between advanced modalities like CT and MRI images has the potential to improve diagnostic workflows and reduce reliance on resource-intensive imaging techniques.

GANs are increasingly central to state-of-the-art machine learning due to their dual capacity for supervised and unsupervised learning. Their generative capabilities have unlocked new possibilities in computer vision, allowing researchers to address challenges previously thought insurmountable. In the context of healthcare, GANs provide a means to tackle critical issues such as data scarcity, modality disparities, and diagnostic automation, all of which are pivotal for advancing edge computing solutions on IoT devices.

2.1.1 Standard GAN

The standard GAN takes random noise as its input (called the latent space as it has the potential to be an image) and outputs a photo realistic image. However, as GANs increased in popularity and ability, they started being used for image-to-image translation called Pix2Pix which led to other types of GANs such as CycleGAN (described below).

GANs work on the concept of a zero-sum game, where the losses of one player are equal to the gains of the other; in the case of GANs, the generator (the blue network in figure 2.1) and the discriminator (the red network in figure 2.1) are trained simultaneously in an adversarial setting. The generator, analogous to a forger, learns to create realistic images, images that are indecipherable from "naturally" created images. On the other hand, the discriminator, analogous to a fraud detective, learns to discriminate between real images and those generated by the generator (the forger)[1].

As training commences, the generator learns to create more realistic-looking images by attempting to fool the discriminator; simultaneously the discriminator learns to detect which images are generated and which aren't. As this battle continues both networks get better at their respective roles until the discriminator can no longer tell the difference between the two types of images, essentially the generator obtains all the points in the zero-sum game leaving the discriminator with a score of zero.

Additionally, since GANs are composed of two networks, they lend themselves intrinsically to

[1]Some of the "real" images could be from another generator not involved with the GAN system being trained but this is rare.

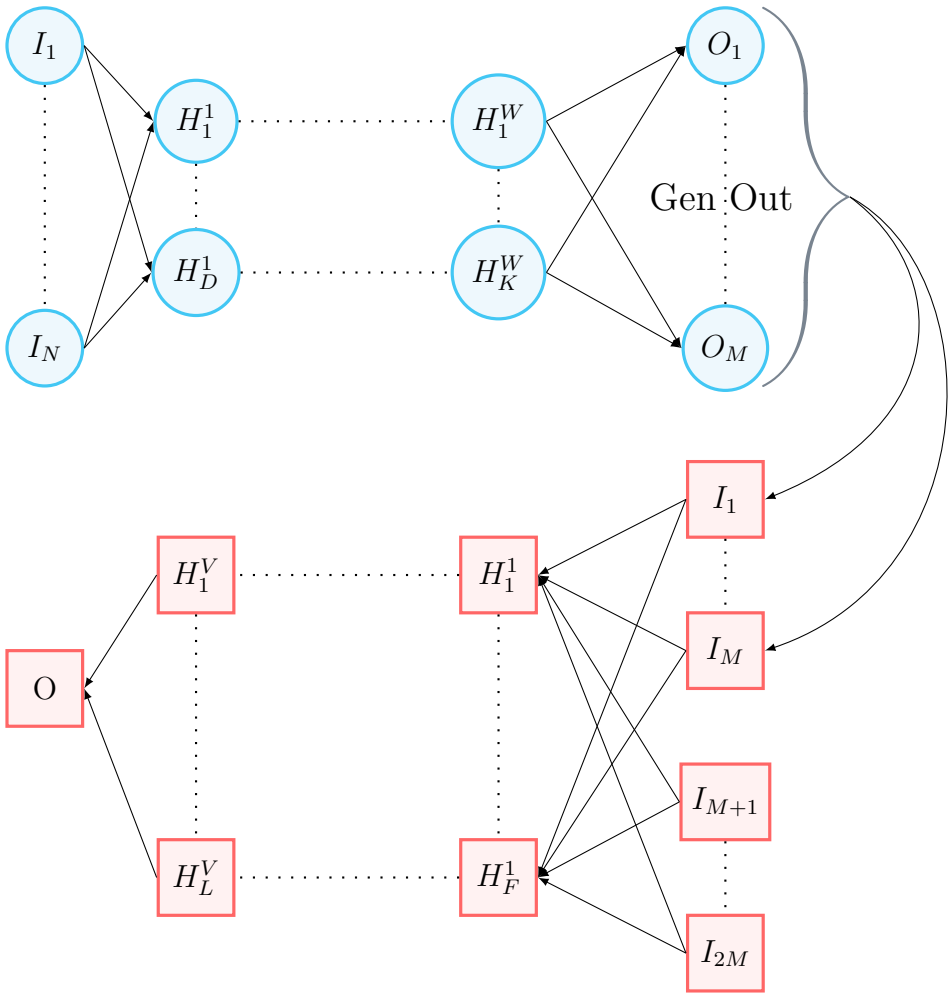


Figure 2.1: GAN Network Architecture. Discriminator inputs I_{M+1} to I_{2M} are for the target image if the network is being used for Pix2Pix, otherwise only a true image or generator output is input to the discriminator and the inputs (I_{M+1} to I_{2M}) aren't used.

distributed deployment, such as IoT, and learning on the edge, which is the largest hurdle in developing on-demand, efficient and practical Machine Learning models.

The following sections delve into advancements in GAN architectures that are most relevant to the research in this thesis. A comparative overview of key GAN architectures is provided in Table 2.1.

2.1.2 Self-Supervised GAN

The increasing prominence of self-supervised learning in machine learning research has extended to Generative Adversarial Networks (GANs) [23], enabling these architectures to benefit from

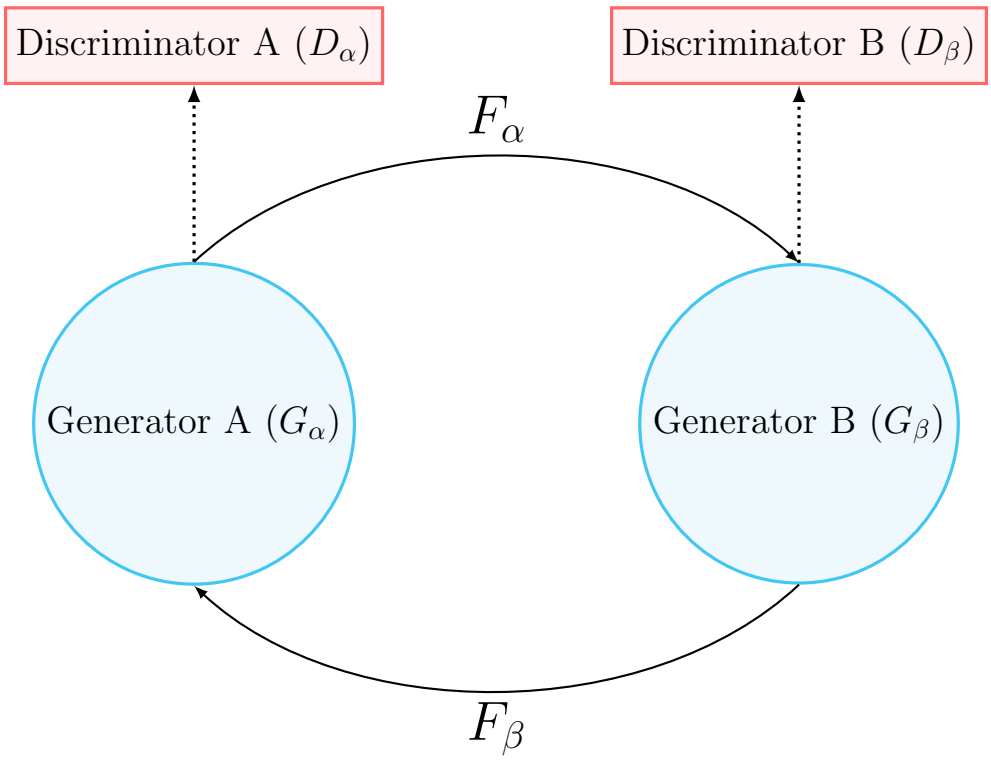


Figure 2.2: CycleGAN Network Architecture. F_α and F_β are functions learnt by Generator A (G_α) and Generator B (G_β) (respectively) to map images from one image domain to another (i.e. F_α maps domain A to domain B and F_β is its inverse).

auxiliary tasks during training. Self-supervised GANs utilise such tasks, such as rotation prediction [17], to train in an unsupervised manner while simultaneously leveraging the advantages of conditional (supervised) models. This hybrid approach not only enhances GAN training but also provides a form of "memory", addressing a key challenge in training on large datasets.

When training a standard unsupervised GAN on massive datasets like ImageNet, whose widely used subset contains 1,281,167 training images, 50,000 validation images, and 100,000 test images spanning 1,000 classes, the model often suffers from "catastrophic forgetting". This phenomenon occurs when the network loses representations of earlier-learned classes, as it focuses on newer data during training. By incorporating an auxiliary task, GAN can establish an additional representation tied to the latent space. This acts as a regularising mechanism, embedding a form of supervision into the network, hence the term "self-supervised".

Self-supervised GANs are particularly relevant to healthcare applications where data scarcity and class imbalance are persistent challenges. The auxiliary tasks can serve as a mechanism for extracting more robust and generalisable feature representations, improving the quality of

synthetic data and enabling better performance in downstream tasks like diagnosis or anomaly detection.

2.1.3 Pix2Pix

Pix2Pix, introduced by Isola et al. [18], is a highly versatile generative adversarial network (GAN) architecture that has served as the foundation for several advanced models, including CycleGAN [19] and the two primary forms of Siamese GANs: disparate image domain transformation [21] and low-resolution to high-resolution reconstruction [20]. Its adaptability and effectiveness have made it a cornerstone in the field of image-to-image translation.

Pix2Pix excels at converting one image domain into another through paired training data, enabling transformations such as labels to facades, black-and-white images to colour, aerial photographs to maps, day-to-night transitions, and edges to photos [18]. Its ability to perform these transformations has catalysed numerous advancements in GAN research and practical applications.

For example, systems like TraVeLGAN [21] extend Pix2Pix's capabilities to translate between unrelated domains. This enables transformations such as mapping endoscopic images to X-ray images, an innovative application with significant implications for healthcare imaging. However, while such outputs may appear visually convincing, they often lack functional accuracy. Despite this limitation, these transformations can still be valuable, particularly when they produce streamlined outputs that enhance the performance of downstream systems. By integrating these results as inputs into later stages of a pipeline, Pix2Pix-based systems can amplify the overall utility and efficiency of the workflow, even in domains like medical imaging where precision is paramount.

Pix2Pix's transformative capabilities and adaptability continue to inspire research and applications, demonstrating its relevance across diverse fields, including healthcare, where it has potential to support innovative solutions in diagnosis, visualisation, and data augmentation.

2.1.3.1 CycleGAN

CycleGAN shifts the classic GAN paradigm without actually changing the core of the GAN system, unlike other GAN variants, such as Pix2Pix and Siamese GAN. Pix2Pix's only difference from a traditional GAN is that the input is a "real" image as opposed to a latent space vector (random noise).

By using two Pix2Pix GANs, CycleGAN learns to map images for one domain into another; this is especially useful for unpaired data as CycleGAN is actually an unsupervised learning technique. As shown in figure 2.2, a pair of generators learn to map to each other's image domain. For example, if the two image domains were alligators (domain A) and crocodiles (domain B), then generator A (G_α) takes images that belong to domain A (images of alligators) and learns to transform them to domain B (images of crocodiles), i.e it learns the function F_α : domain $A \rightarrow$ domain B. On the other hand, generator B (G_β) takes images of crocodiles and learns to transform them into alligators, i.e it learns the function F_β : domain $B \rightarrow$ domain A.

Both discriminators work in much the same way as with a standard GAN, i.e. discriminator A (D_α) learns to differentiate between "real" images of alligators (domain A) and generated images of alligators from crocodiles (G_β(crocodile)). Inversely discriminator B (D_β) learns to differentiate between "real" images from domain B (crocodiles) and generated crocodile images (G_α(alligator)).

However, the utility of the CycleGAN system comes from two additional loss functions: cycle consistency loss and identity loss. Cycle consistency loss forces the two generators to work as a pair; for example, Generator A (G_α) takes an image of an alligator (image X) and returns an image of a crocodile $\hat{Y}$ (such that $\hat{Y} = G_\alpha(X)$). Generator B (G_β) then takes the generated image of a crocodile $\hat{Y}$ and returns a generated image of an alligator $\hat{X}$ (such that $\hat{X} = G_\beta(\hat{Y})$). We therefore have two images, the original alligator X and the generated alligator $\hat{X}$ which is a double generation of the original alligator X (i.e. we have real image X and fake image $\hat{X}$ where $\hat{X} = G_\beta(G_\alpha(X))$). The cycle consistency loss is therefore the difference between these two images as any mapping from domain A to domain B and back again ideally should return

the original domain A image.

Identity loss, on the other hand, is used to ensure that if a "real" image of the target domain is supplied to a generator then the same image, or at least a very similar image, is returned. For example, if Generator A (G_α) is given an image of a crocodile (domain B) it should make no changes as the image already belongs to the target domain.

With all of this together CycleGAN is useful even if the aim is only to map image domains in one direction as it allows for unsupervised learning which is especially prudent in situations where obtaining paired data is impossible as is the case in Chapter 3's research into model to image translation, described in section 3.3.

Currently, the state-of-the-art medical CycleGAN maps paired MRI-CT images on one another to train a segmentation system [24]. While this is promising, it does require paired data which is extremely rare in cross-modality imaging.

2.1.3.2 RL GAN Cooperation

Combining Reinforcement Learning and GANs has an interesting effect. GANs are very capable for image generation tasks [25, 26, 27]; however, when used in real-time, practical applications the training of a GAN is unstable and can suffer from mode collapse [28]. Fortunately, by training a GAN on a latent representation, such as the output of an autoencoder's encoder, the GAN produces more stable results compared to raw input [29]. This approach not only improves convergence but also allows for more robust handling of complex datasets. This can then be taken one step further by inserting a reinforcement learning agent in between the encoder, from the autoencoder, and the GAN. The agent then learns how to optimise the input to the GAN to improve the performance, both the speed and accuracy, of the whole system [22]. However, thus far this technique has been used mostly on reconstruction tasks (3D pointcloud reconstruction, shape completion etc). The combination of Reinforcement Learning (RL) and Generative Adversarial Networks (GANs) has demonstrated promising advancements in addressing key limitations of GANs. Its potential in healthcare applications remains largely

untapped.

For instance, in medical imaging, an RL-enhanced GAN system could be employed for generating high-quality, domain-specific synthetic data from latent representations, addressing data scarcity while maintaining stability and efficiency. Future research could explore how this cooperative framework can be adapted for real-time diagnostic or therapeutic applications on IoT devices.

2.2 Transformers

Transformer networks have emerged as a groundbreaking architecture in machine learning, revolutionising tasks in natural language processing (NLP), computer vision, and beyond. Introduced by Vaswani et al. in 2017, Transformers are based on the concept of self-attention, allowing the model to weigh the importance of different parts of the input data dynamically during processing [30]. Unlike traditional recurrent and convolutional networks, Transformers are highly parallelisable, enabling faster training on large datasets.

The key components of transformer networks consist of:

- **Self-Attention Mechanism:** The core of Transformer networks is the self-attention mechanism, which computes attention scores for all input tokens relative to one another. This allows the model to focus on the most relevant parts of the input sequence, capturing long-range dependencies effectively. The scaled dot-product attention, combined with multi-head attention, enables the model to learn relationships across multiple representation subspaces [30].

- **Positional Encoding:** Unlike recurrent networks, Transformers lack an inherent sense of order in input sequences. Positional encodings are added to the input embeddings to provide information about the relative or absolute position of tokens, crucial for sequential data like text or images [30].

- **Encoder-Decoder Architecture:** The Transformer network typically consists of an encoder-decoder structure. The encoder maps the input sequence into a latent representation, while the decoder uses this representation to generate the output sequence. For tasks like classification, only the encoder may be used [30, 31].

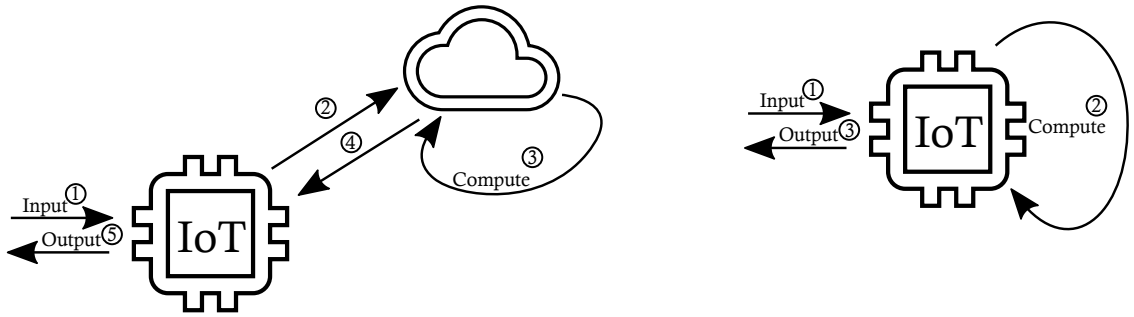
Transformers have shown significant promise in healthcare applications. Vision Transformers (ViTs) adapt the Transformer architecture for image data by dividing an image into patches and processing these patches as tokens. This approach has been applied to tasks, such as medical image segmentation, classification, and anomaly detection, outperforming traditional convolutional neural networks (CNNs) in certain cases [32, 33].

Transformers excel in analysing sequential data, making them suitable for tasks involving Electronic Health Records (EHRs). For instance, models such as BERT have been adapted to extract meaningful patterns from patient records, aiding in diagnosis and treatment planning [34].

While Transformer networks are powerful, they are computationally intensive, requiring significant resources for training and inference. Although Transformers are capable of handling variable-length inputs, enabling distribution across multiple devices, this limitation still poses challenges for deploying Transformers on IoT devices with constrained hardware. Techniques such as model pruning, quantisation, and lightweight Transformer architectures (e.g., MobileBERT or TinyBERT) are active areas of research aimed at making these models more efficient [35].

2.3 Learning on the Edge

Learning on the edge would allow for a great step forward in training bespoke models; by integrating training into the use/deployment of a system two main benefits can be achieved; an edge learning system can use tailored data, which could optimise the models for the targeted task. Additionally, the system could dynamically learn and continue to adapt as it works, since



(a) Traditional Cloud Architecture

(b) On The Edge Architecture

Figure 2.3: Comparison of Traditional Cloud Architecture and Learning on the Edge Architecture

training on the edge would allow the system to learn in real, or near real, time.

To maximise the use of available computing resources, cloud computing is one option for training large models. Given the latency in Internet communications and the advancements in IoT, more and more data analytics systems or applications require real-time responses are built based on Fog and Edge computing architectures. For the purposes of this thesis Fog computing refers to a local server/connection point that is physically close to the user, i.e the user directly connects to the Fog server which reduces latency and doesn't require an internet connection, whereas Edge Computing refers to the computation being run on the device itself or possibly a companion device, i.e a connected smartphone. Research, such as that in [36, 37], has developed an Edge based classifier for seizure detection and compressed sending of results. Whilst this is a great step towards Edge Computing, there is limited Artificial intelligence being leveraged and so these types of systems aren't able to use some of the recent advances in the field of AI for health applications. On the other hand, research such as [38] and frameworks such as TensorFlow Lite [39] take a trained Machine Learning model, in this case Artificial Neural Networks, and convert them into a lightweight form so they can be run on Edge devices. This greatly improves the analytic abilities of Edge devices, but the said models have to be trained on a dedicated system or the cloud, instead of learning on the Edge.

As the IoT field advances and there are increasingly more devices which are more energy efficient, powerful and affordable, the utility of Edge computing is becoming more apparent. Along with reducing the latency issues of cloud computing, the gap between what is feasible in edge learning in practice as opposed to just for research is closing.

Furthermore, Federated Learning (a framework especially suited for private, decentralised datasets) is the perfect bridge between Edge Learning and Artificial Neural Networks, in particular generative adversarial networks (GANs). Federated Learning, described in further detail and its relation to medical image analysis in section 2.3.3, allows for many instances of a model to be trained individually and weight updates are collated and averaged over all instances before the global update is sent to all running models. This allows for improved privacy as the training data never leaves the system and can help train bespoke model instances. GANs provide a fascinating orchestration pattern; since there are two networks in a GAN system, federated learning could allow multi-generators to a single discriminator or visa-versa. This lends itself nicely with Edge Learning as the distributed GANs are then easier to train on edge devices as the number of iterations required is reduced proportionally to the number of federated clients (local training) before all the data is collated on a local server forming a hybrid edge-fog system (for global training). This could then be taken further to produce a, potentially decentralised, system for training a large model.

The following presents an overview of the current state of arts in both Edge Computing and Edge Learning split thematically.

2.3.1 Edge Systems

IoT contains two main architectures, cloud computing and edge computing which differ primarily by the location where the computations take place and additionally differ in computing power. Additionally, there are three primary layers: Cloud, Fog and Mist. [40] These layers are analogous to their namesakes, clouds are large and furthest away from the ground, fog is lighter and hover between the ground and the mist is a thin layer of water molecules suspended just above the ground. The distance from the ground can be thought of as the distance from

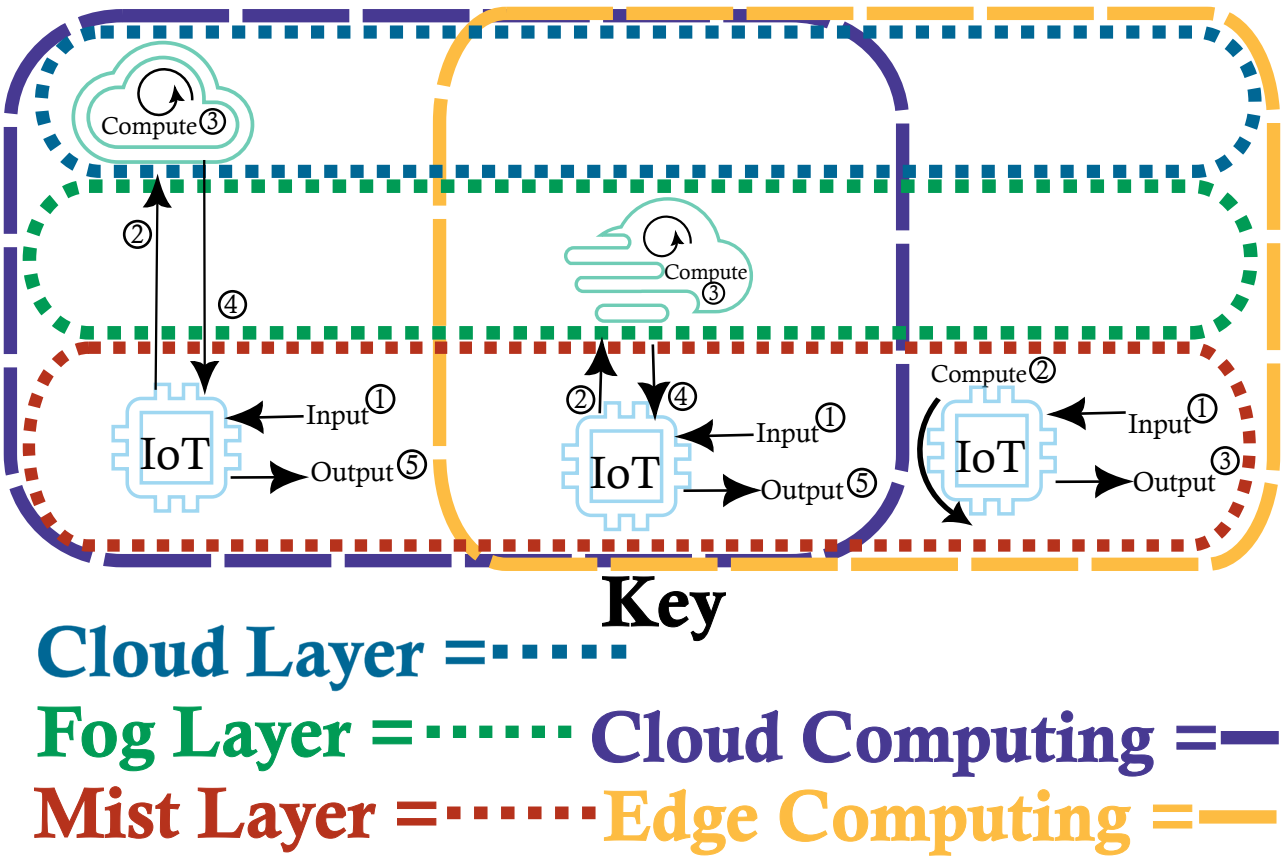


Figure 2.4: IoT Architecture Archetypes: Cloud and Fog both require an external server for handling the computations requested by the IoT device. The primary difference is the server's location, which is local for fog and external for cloud, and communication protocol. Mist, on the other hand, requires no external communication and all computations are done on the device itself.

the users and the size of the water molecules are analogous to the computing resources at each layer.

Cloud computing is still the most prevalent [41, 42] , allowing complex computations to be run on dedicated machines with the results streamed to IoT devices on demand, significantly reducing the need for IoT devices to perform complex calculations and instead act as a go-between for the client and server. This is effective but requires a connection to work; any disruption, due to congestion or network cutout, will cause a delay or even complete loss of functions which is unacceptable in many contexts such as during surgery or within a self driving vehicle.

Fog computing [43, 44] attempts to overcome these issues by exchanging the cloud server for a local server in the same location as the IoT devices; for example, a fog server may be housed

in a hospital, so that every internal IoMT device can connect directly to it, as opposed to an external connection, such as via the internet as is commonly the case in cloud computing. However, fog computing may still experience connection issues and requires space for dedicated hardware in close proximity, which is not ideal for situations where users either cannot be in close proximity or servers cannot be installed in the required location, such is the case with IoT sensors worn on the user who is on the move. Fog computing is a hybrid approach and is there in the intersection of cloud computing and edge computing.

Therefore the most useful archetype is mist computing [23] where all computations are run on the IoT device itself or another (local) participant. Whilst this too requires a connection and is therefore prone to the same issues as fog computing, all participating devices are autonomous and can act individually as required. Unfortunately, this is also the most challenging archetype, as the available resources are heavily limited and often require more specific solutions dependent upon the abilities of the individual device. However, whilst Mist computing itself, is completely within the edge computing boundary it is frequently used in concert with the other layers. Therefore, when referring to edge computing, it is implied that only the mist layer is used, with each IoT device communicating only to each other without the requirement on dedicated (and more powerful) external servers.

Finally, there are additional paradigms that use software approaches to allow for additional benefits; dew computing, for example, caches and mirrors data on the cloud or fog in order to respond as if the data were local but requires syncing with the true cloud servers. The most prevalent examples of dew computing are in cloud file storage where a local computer may contain cached copies of the files which are then synced with the cloud whenever changes are made or a refresh is requested.

Edge and fog computing, as a whole, have made great strides. Mobile Edge Computing (MEC), for example, allows for cloud computing capabilities with vastly reduced latency, high bandwidth and real-time access to network information [45]; therefore, big data analytic techniques can be applied with real-time feedback. However, this system does not actually leverage edge computing's architecture; instead, it uses a fog computing architecture and as such must over-

come issues, such as reliance on users being in proximity to the fog servers, ensuring the connection doesn't interfere with the user's ability to access the internet (mobile data).

In order to overcome these issues, there are a small number of IoT systems that utilize true edge computing; however, these solutions either do not leverage machine learning at all or cannot be trained at the edge. For example, research by A.A. Abdellatif et al. developed an edge based classifier for seizure detection [36, 37]. Whilst this is a great step towards edge computing there is no machine learning being leveraged and so these types of systems aren't able to use some of the recent advances in the field of machine learning for health. On the other hand, IoT devices using machine learning do exist and bringing machine learning, particularly neural networks and their collective family (convolutional, deep, etc), onto the edge is a wide area of research [8]. TensorFlow has the TensorFlow Lite [39, 46] framework which is designed to convert a trained neural network to run on edge devices and is a very popular toolkit and is used in the book, TinyML [47] for running neural networks on embedded devices. However, whilst it can be used for on-device training, to our best knowledge, there are no systems that employ this.

Given the accelerated demand for faster training of more complex neural networks, there has been a focus on dedicated hardware such as GPUs and application-specific integrated circuits, such as Google's TPU (Tensor Processing Unit) and FPGAs (field-programmable gate array), which are inherently cohesive with matrix multiplication operations that underpin the training of neural networks.

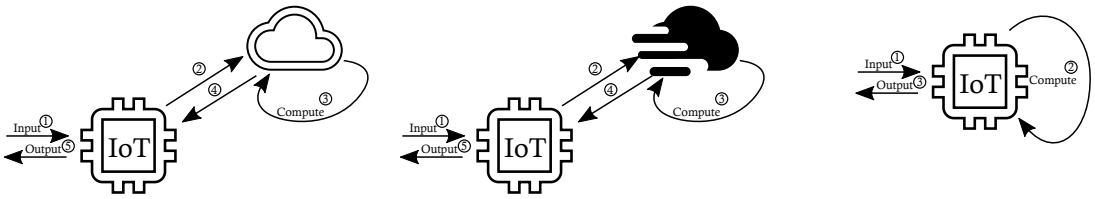
Whilst improvements to these specialized hardware components have accelerated our machine learning abilities, due to their high cost, large footprint, and high energy draw, they are often missing from IoT devices, where smaller and less obtrusive devices are preferred. Even IoT devices that do contain specialized hardware, such as modern mobile phones which may have GPUs, require smaller, embedded, versions that are significantly less powerful than their full-sized counterparts. Furthermore, the less hardware required the lower the baseline battery draw is, a vital factor in IoT.

As a result, learning on a CPU on the Edge is paramount! Removing dependency on specific hardware and providing all benefits via software allows existing devices to use this framework

and keep the footprint of new devices small and focused on efficiency. Intel developed the Intel Xeon Scalable processor which is geared towards learning and inference, focusing on three main features: The compute and memory capacity of the CPU, software optimisations for the CPU for machine learning tasks, specifically DNNs, and the use of distributed training algorithms for supervised deep learning workloads [48]. Although Xeon was designed for servers, these features could be translated to edge computing, for example, the memory capacity allows for better batching of input data and by not using a GPU, there is no overhead between moving data from the CPU to GPU and back again. Also, edge computing and IoT are inherently distributed and as such learning can be offloaded and computed at the source or destination potentially increasing the efficiency of the whole system allowing each device to act autonomously and only interact with other devices when needed [44]. Furthermore, since training occurs upon observation of new data, which may not be consistent, inference, via the previous globally updated network, may be used simultaneously resulting in an online, continuous learning paradigm where networks are incrementally improved while running; this could be further improved by leveraging unsupervised techniques [49] or by using the inferring network as the supervisor.

Therefore, a privacy-ensuring, low-powered and generalizable method is required for training neural networks running at the edge. Federated learning allows multiple networks (of the same architecture) to train simultaneously on non-iid (non-independent and identical distributed) data without ever storing it and aggregating their knowledge into a global network, thereby learning at the edge. However, vanilla federated learning requires a centralised server for the aggregation; when privacy is involved, trust can be an issue. Therefore, a hybrid approach is proposed, utilising the decentralisation of blockchain to create a framework that runs efficiently on multiple IoT devices across a P2P network, removing the need for a trusted centralised server, thereby improving privacy and adding robustness to data integrity. Furthermore, this enables the learning process to be distributed across participating devices (which federated learning does not implicitly do) leveraging the ubiquity of IoT devices to offset their lack of power by increasing the number of devices that are able to run in parallel.

There is a clear hierarchy of architectural archetypes; on one end there is cloud computing



(a) Cloud (Remote Server) (b) Fog (Local Server) (c) On The Edge (No Server)

Figure 2.5: Comparison of Cloud, Fog and Edge Architectures

[41, 42, 50, 51, 52] containing vast resources with increased computational power. This, however, requires communication between the local system and the cloud. Therefore issues such as loss of connection, network congestion, cyber security, etc. will affect the system's performance. On the other end is edge computing [53, 54, 45, 37, 8, 55, 38, 36, 23, 56, 57, 43] with restricted resources but unparalleled access to the device as both the system and the backing computations are on the same (or physically close) device. In between these archetypes is fog computing [57, 43, 58, 59, 44]; having the same structure as cloud computing but instead using a local server as shown in Fig. 2.5. Current solutions target either cloud and fog based computing which do in fact run on the edge; however, they either lack machine learning or only perform inferencing, and not training, which is the most computationally demanding task.

One ideal use of IoMT devices is mobile health (mHealth). In developing countries, this has been shown as an effective method to monitor patients; unfortunately, these systems are often unintelligent relying on basic mobile phone functionalities [50]. These approaches purposefully avoid internet connection, since it is unreliable in many developing countries, yet the required components are in place to leverage the power of machine learning.

On the other hand, the focus of mHealth in developed countries is for smart wearable devices, often paired with a mobile app, yet these too are unintelligent and use a fraction of the ability of modern IoT systems running their computations via the cloud; this allows less capable hardware to run complex computations but suffers from latency issues and must be connected to the internet to work which is not ideal.

Whilst there exists a handful of IoT systems that aim to leverage machine learning on the edge, they only support inferencing and not training.

For example, the STM32CubeAI converts neural networks to run on STM32 Arm Cortex-M-based microcontrollers [60] and has been used to create a human activity recognition (HAR) fitness tracker embedding a convolutional neural network (CNN) in a wrist worn, low power, MCU for inferencing [38]. Frameworks such as these are a step in the right direction but suffer from the need to train the models on a dedicated system or the cloud.

Simultaneously, there have been advances in the hardware required to infer, and potentially train, on the edge. GPUs are better adapted to machine learning methods than CPUs but are rarely found in embedded devices and not all GPUs were created equal; the majority of the frameworks utilise the CUDA language which is designed specifically for Nvidia GPUs. Furthermore, GPUs may be usurped by AI accelerator application-specific integrated circuits such as Google's TPU (Tensor Processing Unit) and FPGAs (field-programmable gate array), which are used in Microsoft's Project Brainwave to improve real-time deep neural network (DNN) inferencing [41, 42]. The requirement for specific hardware increases the physical size, power draw and cost of devices; this is counterproductive for IoMT where smaller and less obtrusive devices are preferred. By moving the learning to the edge on a CPU, one can upgrade existing devices whilst keeping the footprint of newer devices smaller and focus more on efficiency.

The infrastructure required to take IoMT and edge/fog computing to the next level is already in place in a hospital. The users only move within a set area and data collection happens in the same location meaning federated learning is ideally suited to edge learning in a hospital [23]; multiple surgeries happen simultaneously and can all learn together to train models to increase generalisability, improving the model's overall performance by treating each patient or surgery as a decentralised dataset whilst still allowing for bespoke training on a per patient basis [61]. This is ideally suited to clinical settings, as federated learning never shares data thereby keeping data private and allowing training on previously inaccessible tasks such as that of anastomotic leak detection where the existing data, of which there is little, is severely unbalanced.

The majority of Edge Systems for mHealth are still classified as research projects and there are

exceedingly few, if any, commercial systems on the market. However, Edge and Fog computing as a whole have made great strides. Mobile Edge Computing (MEC), for example allows for cloud computing capabilities with vastly reduced latency, high bandwidth and real-time access to network information [45]. This allows for big data techniques to be applied with real-time feedback but comes at the cost of not being a true Edge System but a Fog System (as previously defined) and as such must overcome issues such as reliance on users being in proximity to the Fog Servers, ensuring the connection doesn't interfere with users ability to access the internet (mobile data).

There are also many advancements in Distributed Fog Computing, which has to overcome two main obstacles: the dynamic nature of edge networks and the context-dependant characteristics of the application logic [44]. Not only does this create benefit by allowing tasks to be distributed across multiple devices, it is a natural fit for the distributed nature of IoT and Edge Devices; by allowing each device to run its own system and communicate with others, autonomy amongst the systems is achieved.

On the other hand, there has also been a lot of work into bringing Machine Learning, particularly Artificial Neural Networks and their collective family (Convolutional, Recurrent, etc), onto the Edge. TensorFlow has TensorFlow Lite [39] which converts a trained Artificial Neural Network to run on Edge Devices; it does however, require the TensorFlow Lite interpreter to be run on the devices. It is a very popular toolkit and is used in the book, TinyML [47] for running Artificial Neural Networks on embedded devices. Other software also exists such as STM32CubeAI that can convert Neural Networks, in a particular format, to a C library for use with STM32 Arm Cortex-M-based microcontrollers [60] and has been used by [38] to create a fitness tracker using a Deep Neural Network embedded in a wrist worn MCU.

However, this effort has largely been based on inference as opposed to training. While inference may still be resource intensive for Artificial Neural Networks, the inference stage for other Machine Learning Techniques may not be so intensive, since the goal of Machine Learning is to approximate a function and can often produce efficient algorithms, for example Genetic Algorithms may produce a Boolean formula as its predictor, which can be executed efficiently,

possibly with minor optimisations. The next stage is to bring the learning to the Edge; whilst a difficult step its benefits are unparalleled.

Finally, there have been advances in computing hardware required to infer, and potentially train, on the edge. Whilst GPUs are, in general, better adapted to Machine Learning, specifically for Artificial Neural Networks, because of their high core count, multi-threading ability and are optimised for parallel Matrix manipulation, they are not in every device, although they are becoming more ubiquitous. However, majority of the development frameworks, TensorFlow for example, rely on the use of Nvidia GPUs for acceleration, since only they can use CUDA code, a GPU language specifically for Nvidia GPUs. Furthermore, with the creation of AI accelerator application-specific integrated circuits such as Google's TPU (Tensor Processing Unit), GPUs may not remain the go to hardware for Machine Learning. Finally, FPGAs (field-programmable gate array) allow for hardware changes to be made programmatically to optimise the low-level hardware design and enable parallel processing for specific tasks. Microsoft has leveraged FPGAs in their Project Brainwave in order to improve real-time Deep Neural Network (DNN) inferencing [41, 42]. Whilst all these improvements have accelerated our Machine Learning abilities, they require specific hardware and so are less able to be run on older devices without such hardware and which increase the physical size and cost of said devices. This isn't ideal in IoT, and especially IoMT (Internet of Medical Things), where the smaller and less obtrusive the device the better. This is why learning on a CPU on the Edge is so useful, it can upgrade existing devices whilst keeping the footprint of newer devices smaller and focus more on efficiency.

2.3.2 IoMT devices

Whilst there are many types of IoMT devices, they broadly come under two overarching categories: Sensors (wearable or otherwise) that gather the data and Communication/Collation devices that collect the data (manually or automatically) and send the data to the cloud (see Table 2.2).

The most common mHealth devices used are, multiple or a single, wearable sensors paired to

a smartphone application that connects the devices together and to the network as a sort of broker node that may or may not do some light computation before sending data of to the cloud or to transform the data into a human readable format such as a graph or table. Unfortunately, this system paradigm has many issues such as internet connectivity as there may be network outages or issues may arise in rural or unconnected areas when the devices can't get reliable connections to the internet and as such can't alert users to any issue in time.

2.3.3 Edge learning for medical image analysis

In a hospital setting, it is possible to take IoMT and Edge/Fog computing to the next level. In a large building where users of the system either don't move much or only move within a set area it simplifies the overall architecture. Furthermore, as the use and data collection happen in the same location, federated learning is ideally suited to Edge learning in a hospital [23]; multiple surgeries happen simultaneously and can all learn together to train models to increase generalisability, improving the model's overall performance by treating each patient or surgery as a decentralised dataset, and still allow for bespoke training on a per patient basis [61]. This is especially useful in a medical setting as there are many cases where there just isn't enough available data, as is the case in anastomotic leak detection where there is almost no data and, of the data there is, there's even less with an actual leak. Furthermore, as federated learning never shares the data it is also applicable for sensitive/private data allowing training on previously inaccessible data; with time it may even not require consent as no privacy is violated, although this would require further research as it could be possible to reverse engineer the system to obtain the data [62].

2.4 Federated Blockchain Learning

Federated learning (FL) integrated with blockchain technology has seen a rapid rise in interest, particularly as a means to design privacy-enhancing and trustworthy artificial intelligence (AI) systems, by enabling federated learning to be applied in a decentralised manner due to the im-

mutable and transparent features of blockchain. These systems aim to provide robust security, resilience, and scalability for distributed machine learning tasks.

Yang et al. [63] proposed a federated blockchain framework that employs the Practical Byzantine Fault Tolerance (PBFT) consensus protocol, enabling the system to resist up to 33% of malicious participants while achieving a significantly lower energy cost compared to Proof of Work (PoW). The framework also utilises Multi-Krum as the federated aggregation scheme to ensure Byzantine fault tolerance. However, the framework has limitations: hardware constraints and a lack of real-world testing. The implementation relies on "authorised" edge servers with significantly higher computational power than standard IoT devices. These servers are responsible for validating blockchain transactions, which makes the framework less suitable for low-power IoT devices. Additionally, as far as it is possible to tell, this proposed system has not been tested on real-world hardware but rather on cluster computing systems operating at the fog layer. However, despite this framework not having been real-world tested, it is the current state-of-the-art obtaining over 70% accuracy on CIFAR-10 with up to 20% of the participants being malicious. Unfortunately, this requires over 600 training steps using the AlexNet convolutional neural network (CNN) which has over 60 million parameters and, as already mentioned, is running on a cluster computing system at the fog layer as it would be too large to run on most IoT devices.

In contrast, Islam et al. [64] proposed an innovative approach using drones to facilitate connectivity between IoT devices operating in a pure mist computing environment. Their system integrates differential privacy alongside blockchain-enhanced federated learning to provide an additional layer of privacy protection. However, the scheme requires all participants to pre-register before contributing to the system, posing challenges in dynamic environments where devices frequently join or leave the network.

Both approaches demonstrate the potential of blockchain-enabled federated learning for healthcare IoT systems. Yang et al.'s approach provides insights into energy-efficient consensus mechanisms, while Islam et al. highlights the importance of ensuring robust connectivity and privacy in dynamic edge environments. However, neither fully addresses the unique constraints of

healthcare IoT devices, such as limited computational and energy resources and the real-time requirements for medical data processing and diagnostics.

The integration of federated learning with blockchain in healthcare IoT presents several opportunities for future exploration, most notably the integration of a blockchain consensus protocol, tailored to federated learning on low-power IoT devices. Additionally, given the dynamic nature of surgeries, systems need to accommodate a sudden increase or decrease in participants without adversely affecting the overall system. By addressing these challenges, federated learning with blockchain has the potential to become a cornerstone technology for secure, decentralized, and efficient AI applications in healthcare.

2.5 Differential Privacy

Differential privacy is a robust mathematical framework that protects any single datapoint in the dataset by guaranteeing that the inclusion or exclusion of said datapoint does not, significantly, affect the output of the computation on the dataset, thereby protecting each individual's privacy [65, 66]. This is achieved by introducing controlled randomness or noise to the data or, in the case of machine learning, the model's parameters, preventing an adversary from inferring sensitive information about any individual.

In machine learning, differential privacy is commonly applied during model training to safeguard against privacy breaches in sensitive domains like healthcare. Techniques such as differentially private stochastic gradient descent (DP-SGD) [67] adapt the traditional Stochastic Gradient Descent (SGD) algorithm by including two extra steps: gradient clipping and noise addition. Gradient clipping is applied to the gradient vector ($g(x_i)$) at each step of training to bind the influence of each individual datapoint by scaling the gradient if the ℓ_2 norm is greater than C, the gradient norm bound.

$$\overline{g}(x_i) \begin{cases} \|g(x_i)\|_2 \leq C = g(x_i) \\ \|g(x_i)\|_2 > C = g(x_i)/\frac{\|g(x_i)\|_2}{C} \end{cases}$$

Following gradient clipping, each gradient vector has a noise signal applied before summing and aggregating the gradients into the final gradient vector for that epoch.

2.6 Homomorphic Encryption

Homomorphic encryption allows computations to be performed directly on encrypted data, without requiring decryption, making it highly valuable in privacy-preserving machine learning, where sensitive data can remain encrypted throughout the model training and inference processes, ensuring data confidentiality. While fully holomorphic encryption (FHE) schemes, such as CKKS [68], enable operations such as addition and multiplication it introduces computational overhead, as encrypted operations are typically slower than their plaintext counterparts. Furthermore, matrix multiplication (the backbone of artificial neural networks) requires a large computation overhead [69] and FHE cannot handle non-linear activation functions, such as ReLU, as encrypted values cannot be compared to zero, and requires the use of Self-Learnable Activation Functions (SLAF) [70].

2.7 Functional Encryption

For a given datapoint and a function to be applied to said datapoint, functional encryption allows the owner to encrypt their data and generate a decryption key that will apply the aforementioned function to the encrypted datapoint resulting in the same, plaintext, output that would have been achieved if the original function was applied to the plaintext datapoint [71]. As opposed to FHE there is no need to decrypt the output of the function, however the same issues regarding increased complexity and issues with matrix multiplication still apply. Additionally the owner of the data can place restrictions on the usable functions causing issues of permission and dataset compatability.

Name	Architecture	Uses
Conditional GAN (cGAN)	Conditional label embedded and appended as extra "channel" to input for both Generator and Discriminator.	Constrainable, allowing for specification of output and larger set of generateable outputs. cGAN has a higher accuracy (more realistic) than an unsupervised GAN.
Self-Supervised GAN [17]	An auxiliary task, such as rotation amount detection, is also trained on discriminator.	Allows for cGAN like accuracy without any need for labeling of the data.
Pix2Pix [18]	Unet like architecture, encoder and decoder with skips and possibly additional blocks in between encoder and decoder.	Image to image translation such as segmentation, colorisation, day to night etc.
Cycle GAN [19]	A pair of Pix2Pix GAN's mapping from one image domain to the other with additional constraints on cycle mapping (mapping from domain A to B and back again should result in the input image from domain A).	Transformations from one image domain into a related, though not necessarily aligned, domain; for example, horses to zebras.
Siamese GAN [20, 21]	Using any GAN architecture and adding a siamese network (traditionally used in one-shot learning) into the system in order to group images that belong together in the encoded latent space.	Transformations from completely (visually) unrelated domains such as volcanoes to jack-o-lanterns and clocks to hourglasses.
RL GAN Cooperation [22]	Reinforcement Learning is used to condition the latent space that is input to the generator in order to maximise the accuracy.	3D point cloud reconstruction, increase in system speed for some tasks.

Table 2.1: Comparison of GAN architectures

Type of Device	Data Collected or Given	Use of Collected Data
Mobile Phone/PDA	• Questionnaire/Forms • Video based Information	Data stored on the cloud for analysis by various health authorities (Doctor, Hospital, Healthcare providers etc)
(Non-wearable) Medical Equipment (at mobile site)	• Blood Pressure • Weight • Glucose Levels • etc	Data stored on Cloud for analysis by health authorities
Wearable sensors (Holter monitor, pulse oximeter etc)	• Electrocardiogram (ECG) • Oxygen Saturation • Respiratory Rate • etc	Data sent to and computed on Edge node (e.g. Smartphone, Smart TV, Raspberry Pi etc). Data is intelligently computed before being transmitted to the cloud or directly to health authorities
Edge Devices (Smartphone, Smart TV, Raspberry Pi etc)	• Any data from communicating (potentially wearable) sensor	Intelligently compute data and make decisions about the environment (connectivity/signal, power etc)

Table 2.2: Classes of devices used in IoMT systems

Chapter 3

Enhanced Learning for Gastrointestinal Surgical Interventions

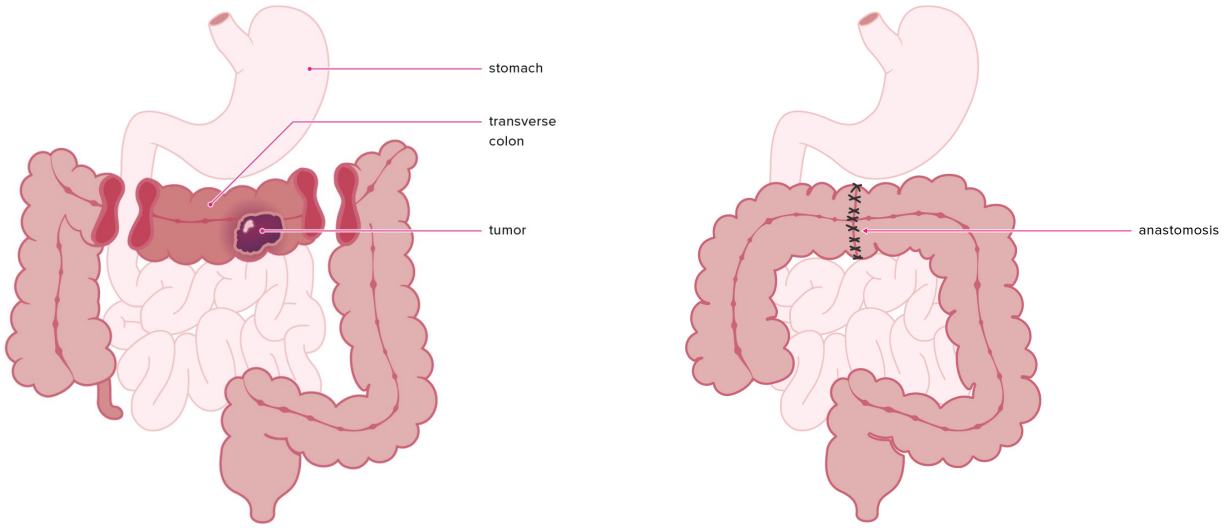
The technical contributions of this chapter are:

- A segmentation system for anastomotic rings taken from arbitrary views.
- Evidence that anastomotic ring bruising is not an effective indicator of a leak.
- A system for mapping 3D models of a "generic" colon to actual images.

Please refer to section 3.1.2 for a full description of the contributions this chapter has made to the field.

3.1 Introduction

In gastrointestinal surgery, bowel resection (Figure 3.1a) may be required for numerous reasons: removing tumours when treating colorectal cancer, familial polyposis, treating genetic diseases such as Crohn's, trauma (such as bullet wounds) and blockages. Once the section of the bowel is removed the intestines require rejoining; intestinal anastomosis is the preferred method and



(a) Bowel Resection (b) Anastomosis

Figure 3.1: Bowel Resection and Anastomosis [1]

is often performed by sewing or stapling the two remaining ends. However there is a risk of anastomotic leak[1], a leak of luminal contents from the surgical join, which can lead to increased lengths of stay for patients, require additional interventions and increase chances of death [72, 73]. Colorectal anastomotic leak rates vary from 6% to 30% depending on various risk factors, this variance in leak rates is also due to the varying definitions of anastomotic leak [72, 74]. It is paramount to identify leaks early to minimise the potential morbidities of this complication. Delays in identifying an Anastomotic Leak has been demonstrated to contribute directly to increased patient risk [75]. Generally, anastomotic leaks are detected via contrast CT scans used to investigate potential leaks, usually after a patient or doctor notices conditions deteriorating. Unfortunately, many of the findings associated with anastomotic leak are neither sensitive nor specific [72]. Therefore, the ability to detect such leaks intraoperatively or even a week later would be an improvement in terms of both patient care and reducing the need for contrast CT's.

However, a colostomy may also be recommended, depending on how much of the bowel was removed, which involves moving one end of the intestine through an opening in the abdominal wall. This in turn is connected to a bag or pouch so that stool, which would normally move

[1]Henceforth anastomosis/anastomotic refers to intestinal anastomosis/anastomotic unless otherwise specified.

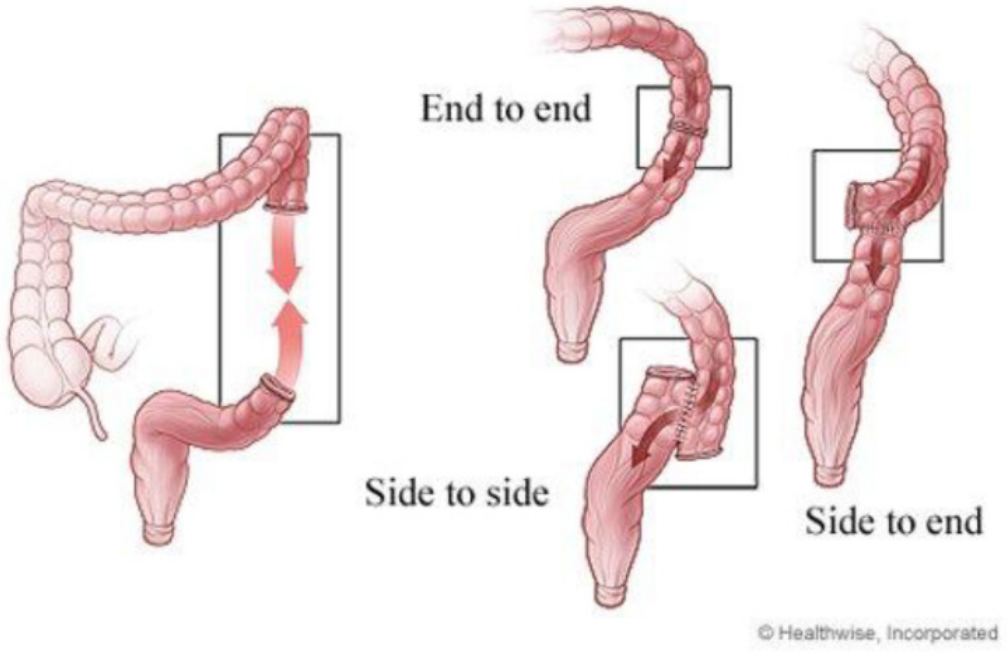


Figure 3.2: Anastomotic reattachment techniques [2] (permission from ©Healthwise, Incorporated. `www.healthwise.org` given to [2]).

through the intestine, instead pass through the opening in the abdomen into the pouch which must be manually emptied out. Ideally a colostomy is used as a short-term solution as it allows the rest of the intestine to rest during recovery; once recovered, an anastomosis is performed, attaching the two sections of intestines back together, Figure 3.1b, with stitches or staples, using one of three techniques: end-to-end (EEA), side-to-side (SSA) and end-to-side (ESA), Figure 3.2.

Although sometimes there's not enough healthy bowel left to do an anastomosis, in which case the colostomy is a permanent solution, the majority of the time an anastomosis is performed and therefore it is imperative to reduce the related risk. In fact, the literature on anastomotic leaks typically has a mortality rate in the 10% to 15% range [76, 77, 78, 79, 80] and a leak can increase the mortality rate from 7.2% (for those without a leak) to 22% [81, 82].

3.1.1 Anastomosis

Anastomotic leak, defined as "a leak of luminal contents from a surgical join", is the most important complication to recognise following gastrointestinal surgery [3]. Any delay in treatment

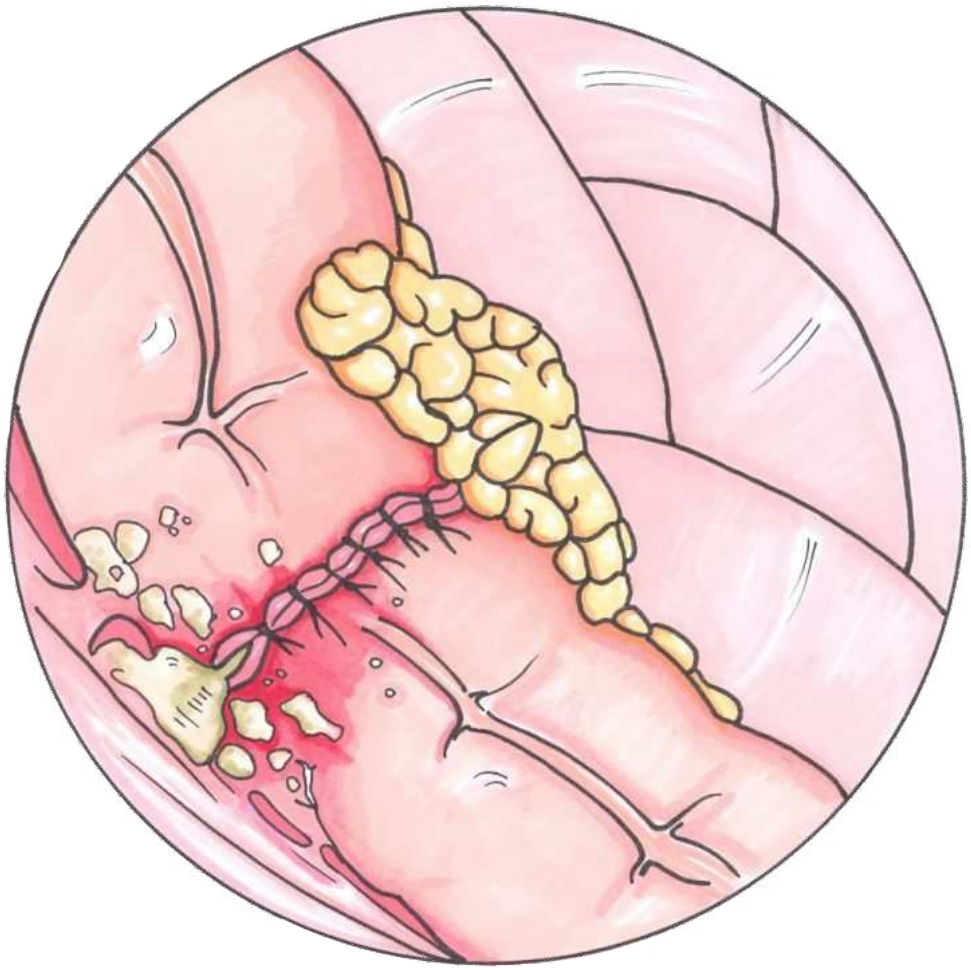


Figure 3.3: Anastomotic Leak. Original image by Aimee Rowe, TeachMeSurgery [CC-BY-NC-ND 4.0] [3]

results in prolonged contamination of the abdomen or chest by the luminal contents, this in turn leads to to the development of severe sepsis and leads to multi organ failure and death. Furthermore, anastomotic leaks have been correlated with an increase in local recurrence and a diminished survival rate after colorectal cancer surgery [76, 83, 84]. Therefore, early diagnosis, resuscitation and treatment of an anastomotic leak is paramount. As a result, for any patient who is not progressing as expected or deteriorates after surgery, the primary diagnoses should be that the patient has an anastomotic leak until proven otherwise.

There are multiple factors that lead to an increased risk of anastomotic leak dependant on both features of the patient and the surgery. Patient factors include “medical” health factors such as existing medication the patient is taking or illnesses, such as diabetes, and their “personal” health factors, such as bodily health (obesity, malnutrition etc) or recreational health (smoking,

alcohol excess etc). On the other hand, surgical factors include the length of the surgery and whether there is any peritoneal contamination (e.g from pus or GI contents). Obviously, the surgeon only has limited control over the surgical factors, and no control over the patient's factors; therefore, a system to detect anastomotic leaks cannot cause an increase in surgery time or any other factors that could contribute to an increased risk of leakage.

Detection of an anastomotic leak usually requires radiologic imaging; in fact, the definitive investigation for a suspected anastomotic leak is a CT scan with contrast of the abdomen and pelvis to determine the presence of any extraluminal contents [3]. Unfortunately, additional investigations, such as blood tests and a clotting screen, may be necessary too. However, the diagnosis may remain elusive or uncertain [76]. A large number of patients, of whom it was ultimately found they had an anastomotic leak, develop a more insidious presentation, often with low-grade fever, prolonged ileus, failure to recover [85], abdominal pain or delirium. In these patients, making the diagnosis may be much more difficult, as the clinical course is often similar to other postoperative infectious complications [76].

3.1.1.1 LumenEye

The LumenEye, figure 3.4, is an endoscope connected to a tablet that allows high-quality imaging of the anorectum at the patient's bedside, reducing the need for formal endoscopy and therefore reducing the cost of diagnosis and removing the need for an operating room when doing a check on a patient.

This is clearly ideal for the verification of anastomotic leaks and an integrated system for such detection would bring immediate benefits to the patient's health, by reducing the need for surgery/radioimaging just to check for a leak, the doctor's time, since the checks can be performed immediately right at the patient's bedside, and the hospital, by reducing the cost of checking and reducing the load on operating rooms and radioimaging machines.

The tablet connected to the LumenEye, figure 3.5, can, although it does not currently, run a machine learning system directly on it which lends itself to two advantages: as the tablet also

Figure 3.4: LumenEye System [4]

has the ability for conference calls for consultations, the proposed detection system could be shared with multiple experts allowing for remote diagnosis and therefore only a technician may be physically required. Secondly, as the proposed system is constrained to be usable on an edge device, when the technology advances, so it could be used with a smartphone or a robotic system, a machine learning algorithm will be well suited to run on such a system.

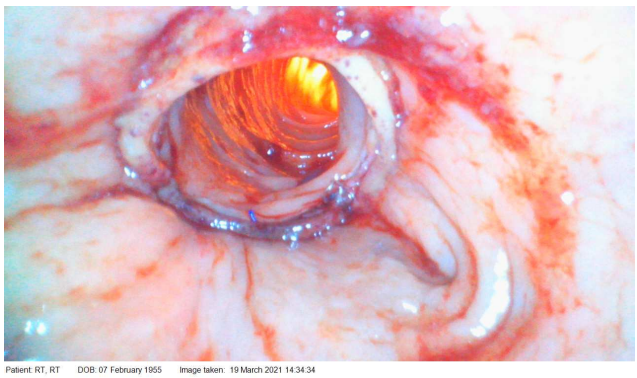
The LumenEye was used by an expert to obtain the dataset containing 7 images (and videos) containing an anastomotic leak and 121 images (and videos) without an anastomotic leak; a subset of this dataset is also used containing 3 intraoperative images containing a leak and 54 intraoperative images without a leak. Figure 3.6 shows examples of each dataset category; unfortunately, it seems likely that white light imaging alone is not enough to detect a leak (especially intraoperatively). In fact many of the images were classified given results found later, for example, the intraoperative leak images were only classified as such following check-ups on the patients the images were taken from, well after the actual image was taken. Therefore,

Figure 3.5: LumenEye Tablet:
`https://surgease.com/our-products`

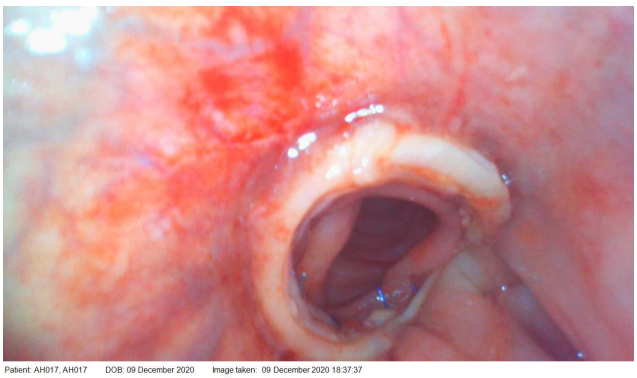
alternative modalities, such as multi-spectral imaging may be required; however, this will need further consideration given the LumenEye's hardware capabilities and the results of further experiments.

3.1.2 Contributions

The research in this chapter intends to allow the detection of an anastomotic leak, without the use of contrast CT scans, by detecting existing or potential leaks viewed by the LumenEye endoscope [4]. Ideally, anastomotic leaks can be detected intraoperatively; however, this is especially difficult as often there are no visually identifiable marks [72] and even potential indicators, such as bruising, may not appear until a few days later. As such a novel method of "day zero" anastomotic leak detection would have a multitude of benefits to patient care and significantly reduce the mortality rate. However, as the LumenEye can be used at a patient's bedside and is simpler and quicker than a contrast CT and/or formal endoscopy, it is still

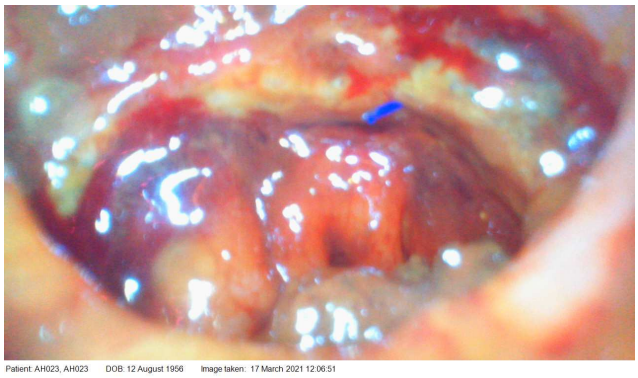


(a) Intraoperative image with a leak



(b) Intraoperative image without a leak

(c) Day 5 image with a leak



(d) Day 5 image without a leak

Figure 3.6: Example images from the used dataset which contains images taken intraoperatively and up to 11 days after the anastomosis was performed.

beneficial even if leaks may only be detectable post-surgery, in fact, most leaks only become apparent 5-7 days after surgery.

This research is based on the hypothesis that the solution to this problem requires a cross-disciplinary approach over existing machine learning techniques; due to the design of the LumenEye, edge computing will have a major influence over the outcomes.

Therefore, this chapter aims to combine state-of-the-art Machine Learning techniques in a novel manner to aid in gastrointestinal surgery targeting anastomotic leak detection and model to image generation. By targeting these two areas, the entire pipeline of gastrointestinal surgery can be improved. Anastomotic leak detection will remove the ambiguity caused by the lack of consensus concerning what defines, and how to diagnose, a leak. On the other hand, model to image generation can be used for both alternative imaging and generating bespoke patient data for training these detection networks or systems used later in the pipeline. Additionally,

anastomotic leak detection is required for effective post surgical care to minimising the risk of complications.

With a framework like this in place, one possible evolution of this research could be in autonomous surgical robots where the surgery can then be performed by training an autonomous system with imitation learning using the bespoke generated patient data and using augmented reality (AR) to help guide the surgeon in tandem with said system.

This chapter discusses the current techniques that could be adapted for this task, the challenges and what a successful outcome may look like.

From a technical standpoint, the main contributions of this research are:

- A segmentation system for anastomotic rings taken from arbitrary views.
- Evidence that anastomotic ring bruising is not an effective indicator of a leak.
- A system for mapping 3D models of a "generic" colon to actual images.

3.2 Segmentation Network - Ring Segmentation

3.2.1 Problem Overview

The first step into anastomotic leak detection was to create an anastomosis ring detection network using a U-Net architecture, as this has been used successfully in many medical imaging tasks [11], with Residual Bottleneck Bridging layers [86]. Other methods, such as One-shot, learning were considered but ultimately decided against since anastomoses are not necessarily similar to one another and the variance in image angles was too wide. Self-supervised learning, on the other hand, could be used in the future and has shown impressive results in order to pre-train the encoder sub-model for a segmentation system [87]; however, the obtained results so far are accurate enough that this is unnecessary at this time. Furthermore, the self-supervised task used in [87] was bespoke to the task (kidney side prediction as the self-supervised task

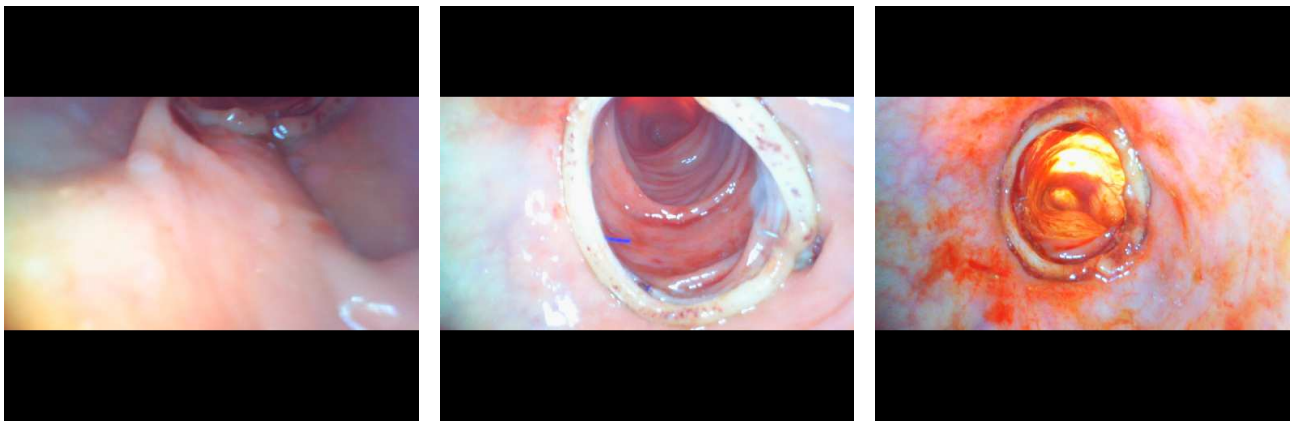


Figure 3.7: A subset of our Anastomosis Images dataset

for kidney segmentation) whereas, for the task of anastomosis segmentation, any auxiliary task would be somewhat generic, for example, rotation amount detection, and as such self-supervision is not the correct fit for this system. The data used for training this model is discussed in the following subsection (section 3.2.1.1), including how it was generated and by whom. Unfortunately, anastomotic leak detection is especially difficult as often there are no visually identifiable marks; although there are some potential indicators, such as bruising, these may not appear until a few days later. Furthermore, there are two major hurdles to building an integrated system to detect leaks intraoperatively; the availability of "day zero" images taken after stapling (section 3.2.1.2) and the fact that often there are no indicators visible immediately following surgery (section 3.2.1.3), it can even be impossible to visually identify a leak. It is these challenges that lead to a divergence from a standard classification task to a series of systems; starting with this segmentation network in order to get a closer look at potentially interesting sections of the image, primarily the anastomotic ring where the joint occurs. This is then followed by image analysis techniques, such as K-Means, as a means of semantically understanding the anastomosis and using this as a predictor of a leak.

3.2.1.1 The Data

Since the purpose of segmenting the ring is to aid in detecting a leak, the dataset comes from the same set that will be used for leak detection; these images of anastomosis were taken with the LumenEye by qualified practitioners, as would be the case in practice (fig 3.7).

Figure 3.8: Corresponding masks for each Anastomosis image above

However, as this dataset did not include a masked segmentation for the ring, the corresponding segmentation masks were manually created in Krita [88] by a non-medical practitioner (myself) by painting over each dataset image (as shown in fig 3.8); whilst this isn't exactly an expertly segmented ground truth, it is accurate enough to train the system for the required purpose and can also be tailored to the task if a part of the pipeline further downstream is discovered that could be improved by changing said masks. The images in the dataset were taken at various points falling into one of two categories, intraoperatively and 0-5 days after the anastomosis was performed.

3.2.1.2 Lack of "day zero" images

Typically machine learning requires large amounts of training data to be useful compared to more traditional methods. This requirement is increased for deep networks and it is widely recognised that successful training requires annotated training samples in the thousands [11]. However, the available dataset contained a meagre 56 images of anastomosis, only 3 of which resulted in a leak. This caused two problems, primarily the lack of images but also that a standard classification network was never going to work; in general, most networks tried scored over 94% accuracy by just always classifying the image as "no leak".

Deep convolutional networks have outperformed the state of the art in many tasks [89, 90] and are the most likely candidates to give good results. While convolutional networks aren't exactly new [91], as mentioned already they are highly dependant upon the size of the training data;

for example, the work by Krizhevsky et al. [90] used the ImageNet dataset which contains one million training images. As such, the small number of total images in the dataset used in this chapter was a problem; although in general, deeper networks are superior to shallower networks, increased depth means a larger number of network parameters which is more prone to overfitting and is compounded by the limited number of training examples[92].

In order to combat this a U-net like architecture with residual bottleneck layers was implemented as described in [11, 86, 24]. This allows the training of a deep network regardless of the limited dataset, made even more limited by the fact that the dataset required splitting into 36 images for training and 20 for testing. The U-net architecture includes an encoder and decoder that allows the network to use both localisation and, due to the skip connections, use the context of the overall input to generate a highly accurate segmentation.

3.2.1.3 No visible indicators

As ideal as it would be, artificial neural networks are not magic, they cannot tell us something if it is not there and as such, a leak detection system has the same issue that a doctor, looking at the same image has, if there are no visual indicators then nothing can be predicted. However, neural networks can do something humans cannot, they can look at all the data at once whereas humans often focus on a subset; furthermore, neural networks can view the full image at the pixel level which is infeasible for a person. Therefore, the data was reduced to include only the areas of immediate interest, in this case the anastomotic ring (where the joint is), and from there look for any indicators; for example, bruising is a potential indicator of a leak occurring and so, by reducing an image to contain only the relevant areas, the network can then limit its search to look for indicators of a leak. However, as mentioned in subsection 3.2.1.2 only a small number of images (less than 6%) actually resulted in a leak. Therefore, one avenue could be to apply clustering techniques such as k-nearest neighbours (kNN) to analyse the data and look for patterns that may not be apparent to the human eye enabling classification. This lends itself nicely to an online learning structure which also helps combat the lack of training examples. This challenge is, however, at the moment essentially unbeatable; if the data is not

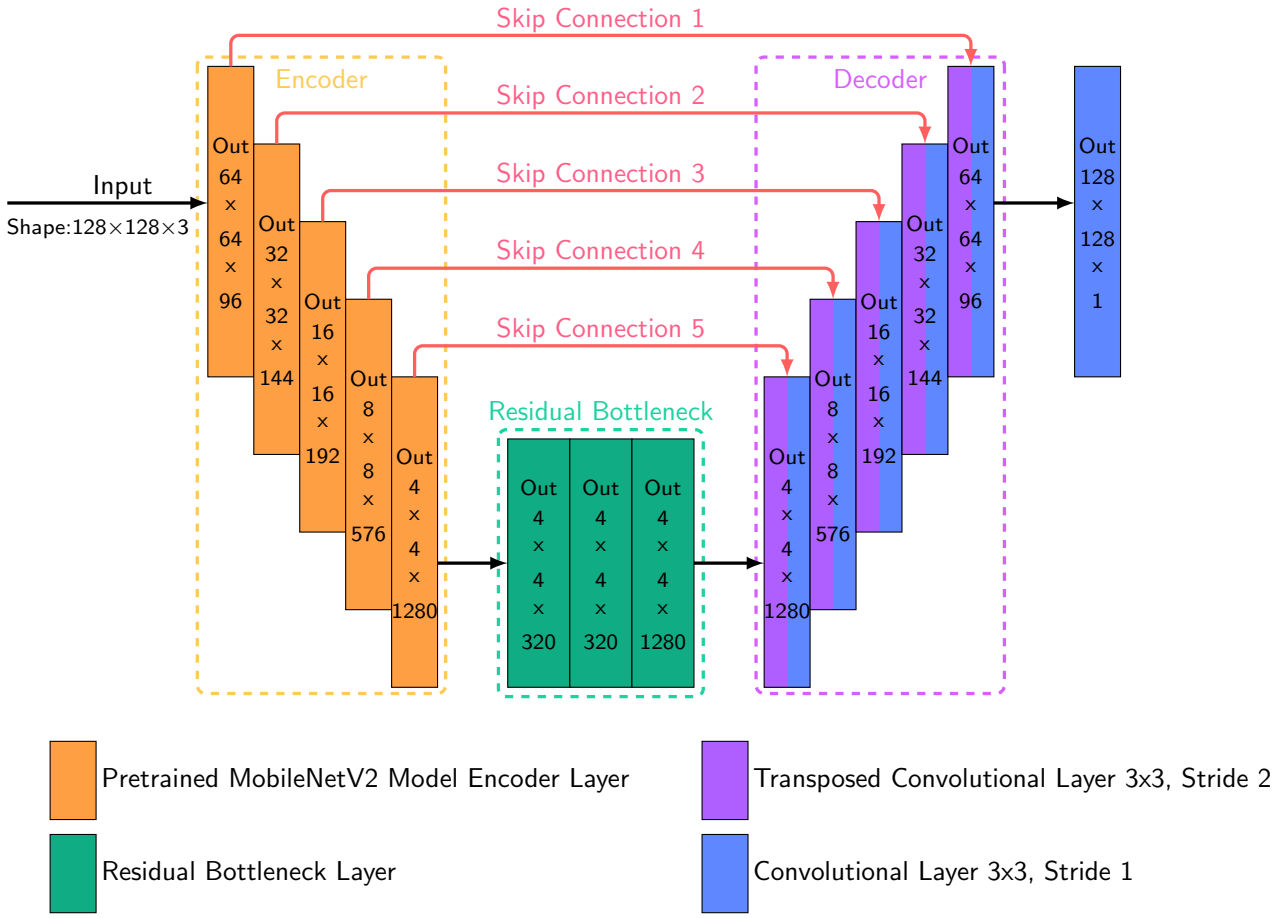


Figure 3.9: The Segmentation Network Architecture

there then it is not there but attempts can be made to mitigate its impact.

3.2.2 Ring Segmentation Methods

For the objective of designing a system that can detect an anastomotic leak or ideally predict that one will occur after an anastomosis is performed, a segmentation network was built first to be able to extract the region of interest (roi) that a doctor would typically look at to determine if there was the potential for a leak. As the model was built with a U-net like architecture there are three main parts of the network. The encoder was formed from pre-trained layers from the MobileNetv2 model; while this model was trained for classification on the ImageNet dataset, containing 10,000,000 labelled images belonging to over 10,000 categories, only the first few layers were used as these usually learn far more general features and was therefore appropriate to use in this segmentation network. These layers are shown by the orange blocks

in figure 3.9 and are described in subsection 3.2.2.1. This is followed by the residual bottleneck layer that uses identity mappings as the skip connections and after-addition activation, based on the work described in [86] and used in [24], explained in subsection 3.2.2.2. Finally, the decoder resizes the image back to its original size and is followed by a final convolution, which whilst not technically part of the decoder is included in the decoder's explanation in subsection 3.2.2.3.

3.2.2.1 Encoder Block

Each layer in the encoder downsamples the image to half of the previous layer and adds an increasing number of filters; the segmentation model takes, as its input, colour images containing three channels (red, green and blue) of 128×128 pixels with the encoder block transforming the input image as follows:

$$\overbrace{128\times128\times3}^{\text{input size}} \rightarrow \overbrace{64\times64\times96}^{\text{layer 1 output size}} \rightarrow \overbrace{32\times32\times144}^{\text{layer 2 output size}} \rightarrow \overbrace{16\times16\times192}^{\text{layer 3 output size}} \rightarrow \overbrace{8\times8\times576}^{\text{layer 4 output size}} \rightarrow \overbrace{4\times4\times1280}^{\text{layer 5 output size}}$$

Using pre-trained layers has two primary benefits in this case, most obviously it is no longer necessary to retrain the layers but also that they are more generalised as they have been trained on a more varied input; whilst the destination dataset is homogeneous and very different from anything in the pre-train dataset, overfitting is a serious concern due to the small size of the dataset and the depth of the network. However, as a consequence of using an off-the-shelf pre-trained network the architecture is not customisable and therefore the encoder must also have 5 layers as these layers are taken directly from the pre-trained network.

3.2.2.2 Residual Block

Three residual blocks are sandwiched between the encoder and decoder layers and are designed as described in [86] where, instead of applying normalisation and activation after multiplying the input by the weights, they are pre-applied, albeit in the same order so as not to mutate the

identity when it is added back into the output. These blocks improve the non-linear capability of the network by adding the identity function back into the network. The residual layers are actually a combination of three: A feature-down convolution, which downsamples the input's filters (in this network to $\frac{1}{4}$ the number of filters) using a kernel of size 1. This is followed by a standard convolutional layer, which is now operating on a smaller input and as such keeps performance up. Finally, a feature up convolution is used which is the exact opposite of the first layer (the feature-down convolution) and returns the number of filters back to the original input. After all residual layers are used a rectified linear unit (ReLU) activation layer is used.

3.2.2.3 Decoder Block

The decoder block is symmetric to the encoder block and is built with pairs of layers, the first layer in the pair is a transposed convolution that doubles the size of the input with the same number of filters as the encoder layer at the same depth. The second layer in the pair is a standard convolution with the same number of filters as the first; both layers are followed by instance normalisation [93] and ReLU activation [94].

Included in the decoder block, although not actually a part of the decoder, is a final convolutional layer that uses tanh activation to normalise the output [24] which is an image the same size as the original input (128×128) but with only two channels representing the probabilities of each pixel belonging to the background or the roi.

Similar to U-Net [11], long-range shortcuts connecting the encoder and decoder layers at the same depth are used which smooths the results and gains an increase in convergence speed [24].

3.2.3 Results

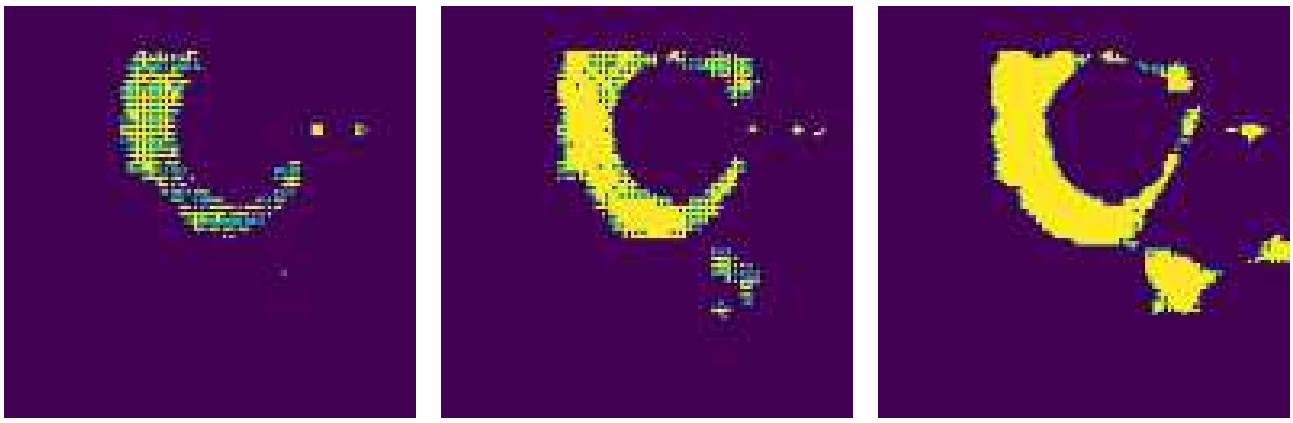
Multiple variations based on the final network were tested, with all networks using the Mean Squared Error (MSE) loss function: The first network was a simplified version using the same layers but connected differently such that the last encoder fed straight into the residual blocks

(a) Input Image (b) Ground Truth

Figure 3.10: Input and Ground Truth segmented ring. In the ground truth image and subsequent output images the purple pixels are used for class 0 (the background) and the yellow pixels belong to class 1 (the foreground). However, there are also green pixels caused by interpolation when upscaling the outputs for display; these pixels have been left in as they are a good qualitative show of uncertainty and borders. Note that all the network segmentations in the following figures are predicting this ground truth image.

without a skip connection, while the previous skips remained. This resulted in asymmetry in the encoder and decoders using five encoders and four decoders with the final layer as a transposed convolution with stride 2. This model trained faster but needed more iterations with an accuracy between 94 and 95%. The results after 70, 100 and 150 epochs are shown in figure 3.11 with the losses shown in figure 3.13a.

Following this, the network was finalised as described in section 3.2.2 and the weighting for each class was tweaked. Ordinarily, this is used to include a weighting to the loss function since this image segmentation, like most others, is treated as a per-pixel classification task and consequentially there is an imbalance between the number of background and foreground pixels; however, in this case, misclassification of the background was penalised as it resulted in a more accurate and conservative segmentation of the anastomotic ring. A weight of 3.075 (penalising the misclassification of the background by just over three times the misclassification of the anastomotic ring) provided the best balance of accuracy and precision.



(a) 70 Epochs, Dice Score: 87.54%

(b) 100 Epochs, Dice Score: 83.87%

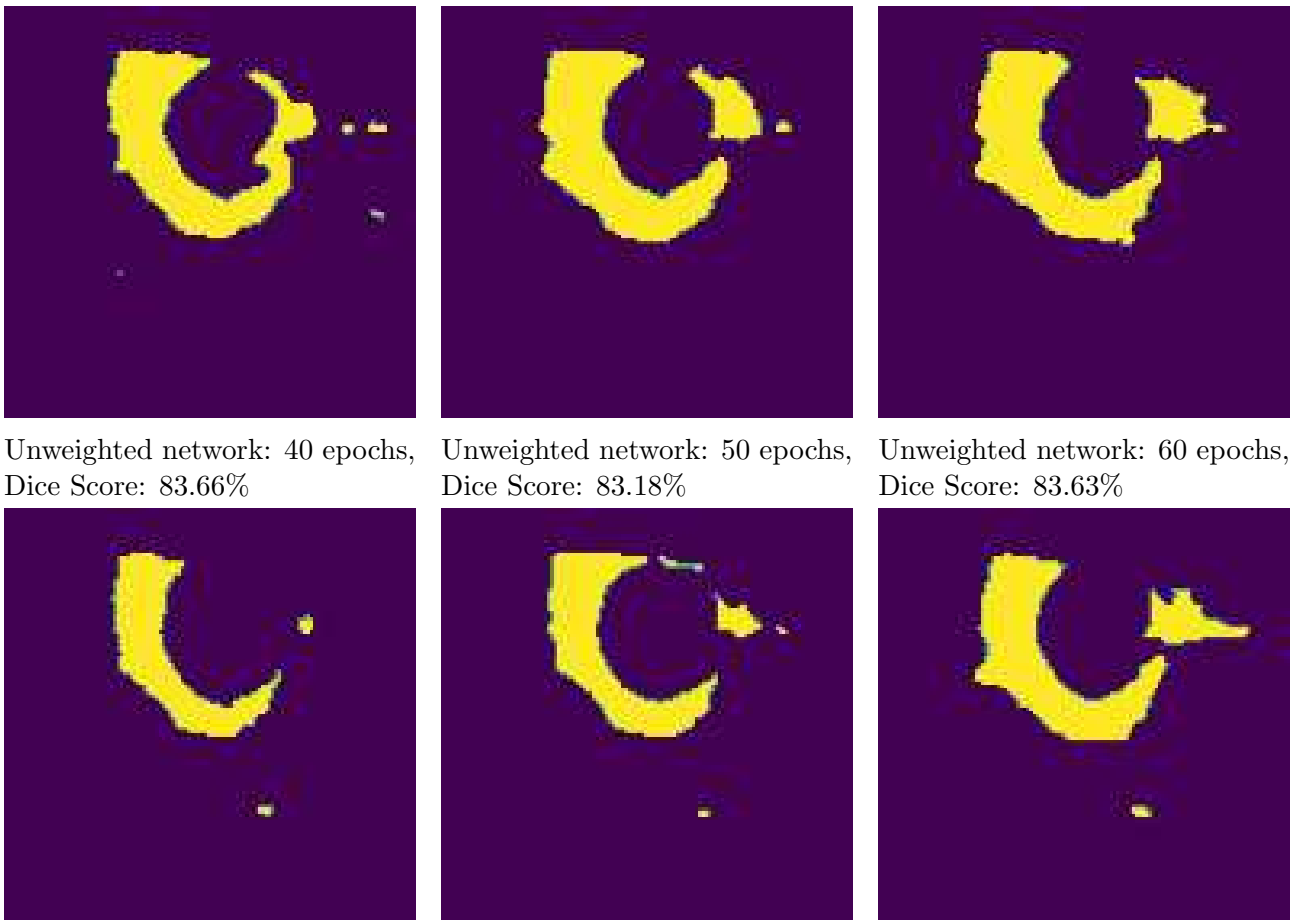
(c) 150 Epochs, Dice Score: 81.94%

Figure 3.11: Example outputs from the simplified network after the corresponding number of epochs

The epochs for the unweighted network are shown in figure 3.12 with the corresponding losses shown in figure 3.13b. This demonstrates that the final network has significantly improved upon the simplified version and requires significantly fewer epochs, however, it overestimates the number of pixels in the ring. On the other hand, the weighted network uses a ratio of 3.075 to 1 for classifying the background to the ring; this requires more iterations as the network is more conservative but results in a higher accuracy. These results are shown in figure 3.12 with the losses shown in 3.13c.

3.2.4 Bruising Percentage

One noticeable feature common to the images in the dataset was bruising; therefore, by detecting the amount of bruising present on the anastomosis, it was hypothesised that this could be an indicator of whether or not a leak had occurred. Starting with the same network used to segment the anastomosis (as described in section 3.2) and applying some small modifications such as removing the loss weighting as the bruising area was significantly smaller than the anastomosis alongside re-training using a different subset of anastomosis images that contained visible bruising, this modified network was used to obtain the anastomotic ring. The ring's bruising was then extracted using the hue saturation and value (HSV) as a guide, this was chosen heuristically and optimised using a subset of the input dataset to contain all the pixels



Unweighted network: 40 epochs, Dice Score: 83.66%

Unweighted network: 50 epochs, Dice Score: 83.18%

Unweighted network: 60 epochs, Dice Score: 83.63%

Weighted network: 80 epochs, Dice Score: 86.96%

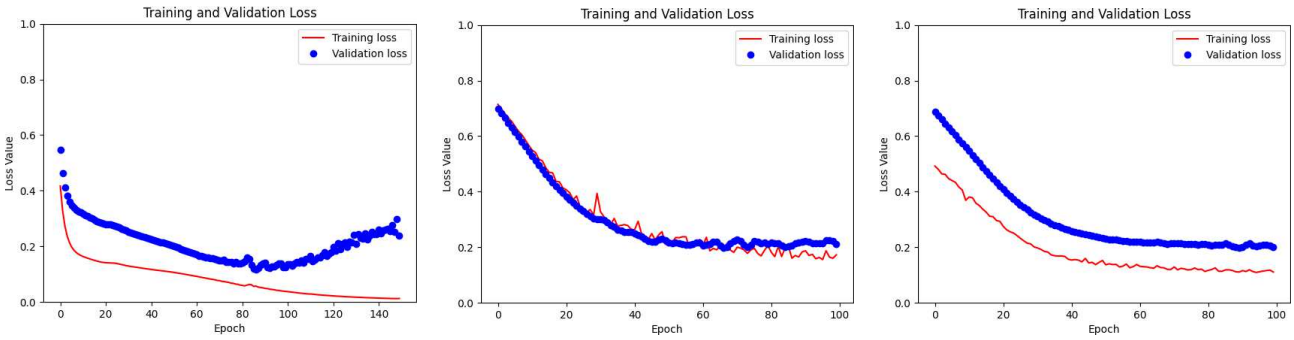
Weighted network: 90 epochs, Dice Score: 85.71%

Weighted network: 100 epochs, Dice Score: 84.04%

Figure 3.12: Example outputs from the final network after the corresponding number of epochs, the top row contains the results from the unweighted network and the bottom row displays the results from the weighted network penalising misclassification of the background. Note that the unweighted network required less epochs for similar results to the weighted network.

with a HSV between (110, 12, 0) and (161, 40, 255). Finally, the percentage of the anastomosis that contained bruising was calculated on a pixel-wise basis. Figures 3.14 and 3.15 show some of the results using one of two models, one trained solely on intraoperative images (figure 3.14) and one trained on images taken 0 to 5 days after the surgery (figure 3.15).

Table 3.1 shows the mean, median and range of bruising percentages for each of the models and whether or not a leak occurred in the set of images provided. Despite the large disparity in the number of leak images vs non-leak images, there was still too little variance to allow a conclusive result on whether or not bruising can be used as an accurate indicator for an anastomotic leak.



(a) Simplified network loss (b) Unweighted network loss (c) Weighted network loss

Figure 3.13: Loss graphs for the simplified network and both the unweighted and weighted versions of the final network. The red line is the training loss with the validation loss shown in blue. The loss used is the Mean Squared Error (MSE) and the weighted network loss uses a weight of 3.075 for background classification penalising misclassification.

Model	Leak			No Leak		
	Mean	Median	Statistical Range	Mean	Median	Statistical Range
Intraoperative	9.87	12.07	11.55	8.65	7.20	32.35
Day 0-5	7.46	9.55	9.53	7.22	5.49	28.13
Day 0-5 Improved	8.02	7.06	10.51	7.15	6.20	27.38

Table 3.1: Bruising Percentage. The statistical range is defined as the maximum value observed - the minimum value observed

3.2.5 Conclusion

A fully connected, convolution-only U-Net-inspired architecture with residual layers enables highly accurate segmentation, even with a limited number of training examples despite no particular techniques being employed to specifically target limited datasets. However, bruising on the anastomotic joint has been shown to be an unreliable indicator of leaks. Combined with the restricted viewing angles of the anastomosis, this limitation imposes a significant challenge on the current system's diagnostic capabilities. Whilst there is a lot of research on anastomotic leak diagnosis, as far as it is possible to tell, there is no research into machine learning solutions. As such, there is no state-of-the-art to compare to; however, comparing the segmentation system with polyps state-of-the-art segmentation techniques, which obtains a top dice score of 85.9% for SD images and 92.9% for HD images [95], shows that the anastomotic leak segmentation system in this chapter is sufficiently accurate.

However, if instead of images the dataset contained a series of slices of images, this would allow

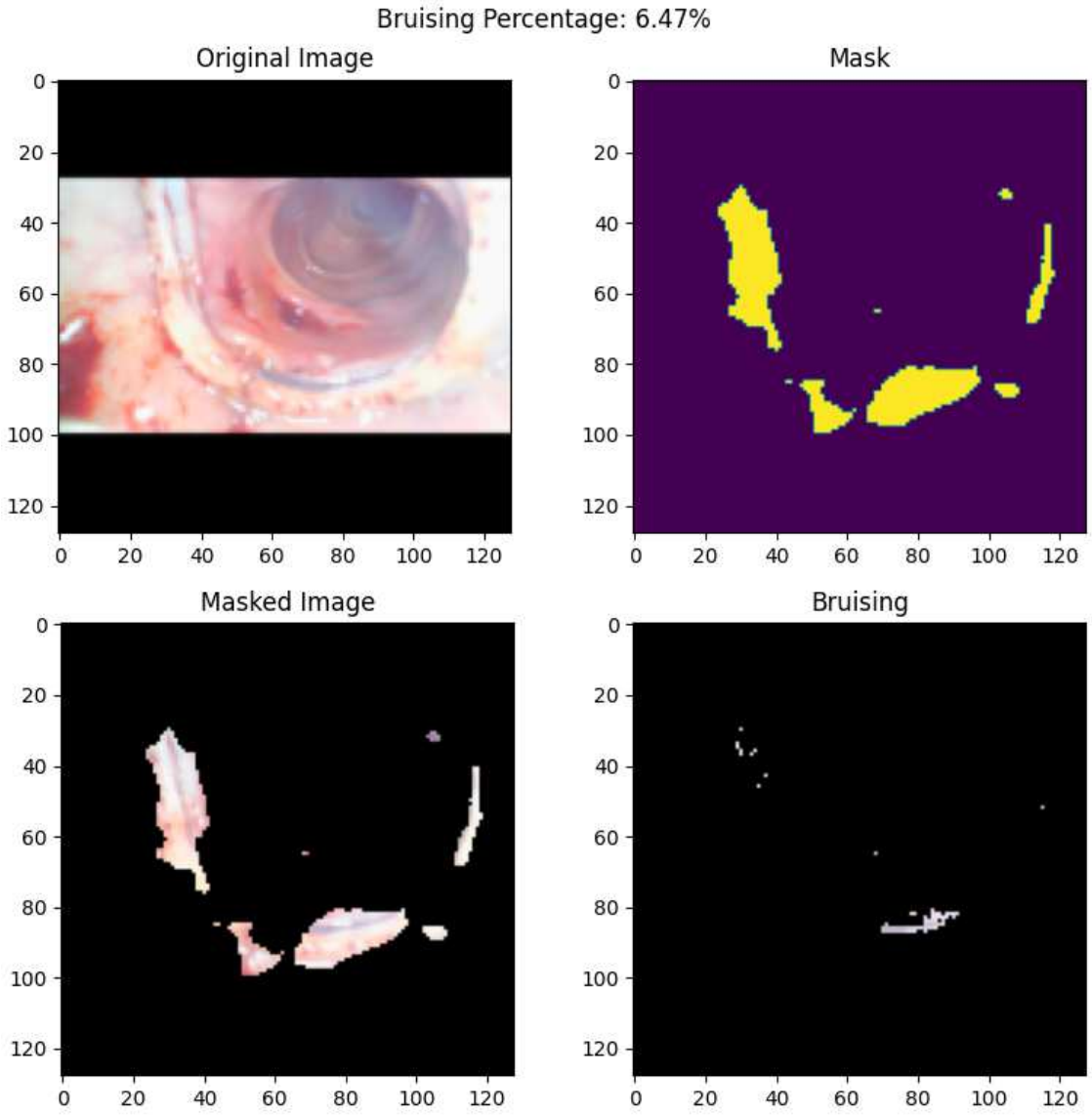
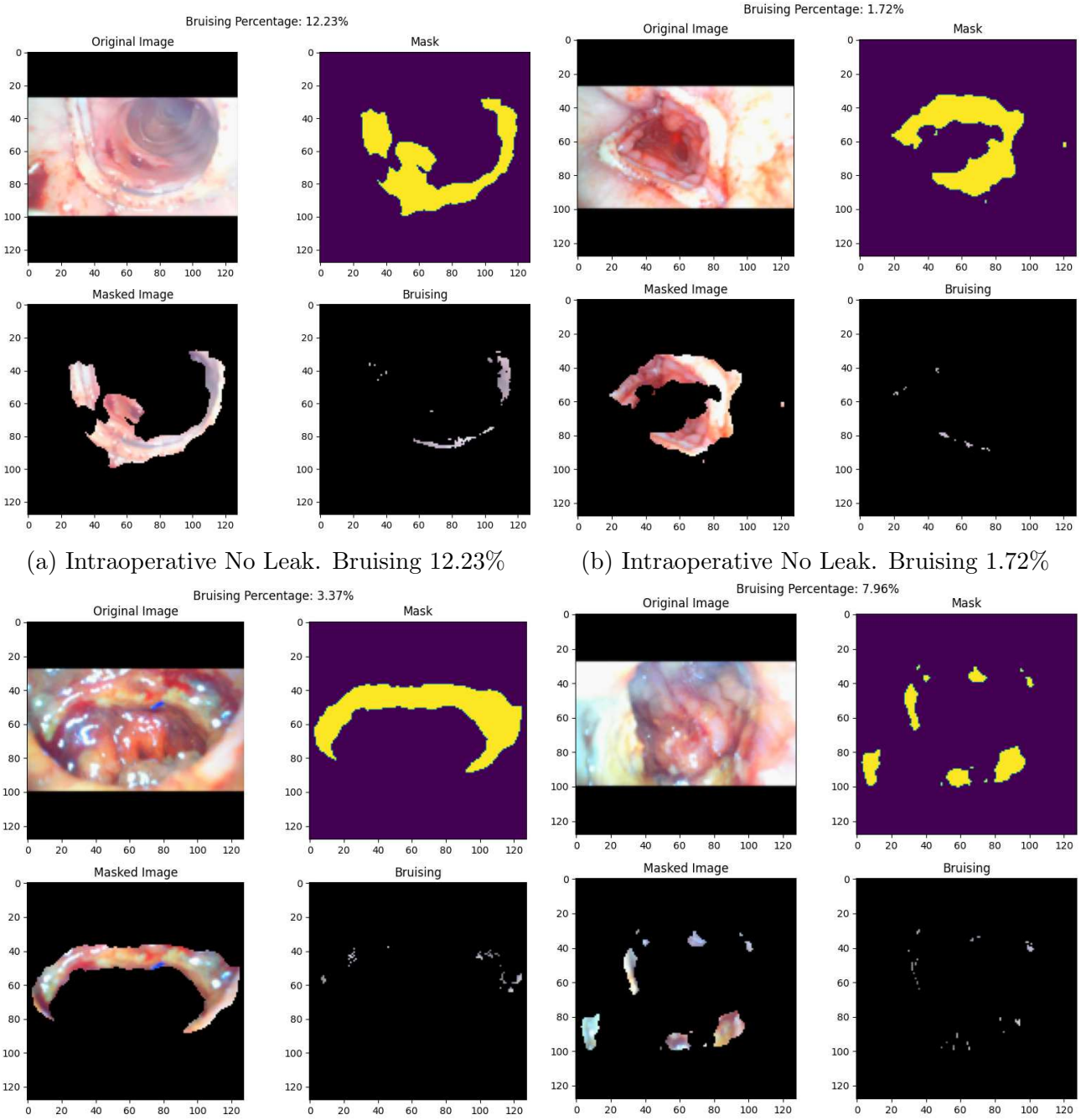


Figure 3.14: Bruising percentage calculated using the intraoperative network (trained solely on intraoperative images) on an intraoperative anastomosis which resulted in no leak

for a three-dimensional view of the anastomosis aiding in the segmentation and identification of leaks. Additionally, incorporating an audio recording of the water pump test, commonly used to detect potential leaks, could further enhance the network by providing an additional physical data source. This supplementary input may capture subtle acoustic patterns that are imperceptible to the human ear, potentially offering valuable insights to augment the network's decision-making capabilities.

This is only the first step in detecting or predicting anastomotic leaks and much more research is needed to build a fully functional system. Potential avenues lie in using generative adver-



(a) Intraoperative No Leak. Bruising 12.23%

(b) Intraoperative No Leak. Bruising 1.72%

(c) Day 5 No Leak. Bruising 3.37%

(d) Day 5 Leak. Bruising 7.96%

Figure 3.15: Bruising percentage calculated using the day 0 to 5 network (trained on images taken 0 to 5 days after the surgery)

sarial networks to generate more training images or reinforcement learning for learning what additional features are indicative of a leak that may not be known or visible to doctors.

3.3 Model To Image Translation

Given the conclusion of the research into anastomotic leak detection discussed in the previous section (3.2.5), taking the route of generating additional training images is a logical next step as it connects perfectly with two previously mentioned concepts. Firstly, additional views of the data would help with both the scarcity of data and the lack of multiple angles of each anastomosis; this can be used for both alternative imaging and generating bespoke patient data for training the new generation of intelligent imaging systems. Secondly, generating images is the realm of GANs and, as previously mentioned, GANs lend themselves intrinsically to edge computing which further incentivises the use of IoT devices in a hospital setting.

There is currently insufficient data to train generic (and practical) models for analysing endoscopic images; however, by artificially generating this data instead from a generic 3D model, required data could be obtained on which AI models could be trained. These systems could then be trained as an alternative to traditional imaging modalities such as MRI, CT, and ultrasound where they may not be applicable or feasible. Furthermore, the generated data could be patient-specific allowing for the training of a bespoke system to then aid in surgery which is invaluable as data collection can be difficult, time-consuming and may not even be specific enough. Using a system such as this, artificial issues such as polyps and other tumours can be generated which enables the training of better models as the exact data necessary to train the optimal model can be obtained, removing the dependency on dataset collection. Furthermore, by using endoscopic images from an actual patient, one could then generate patient-specific data which can be used to train a bespoke model to be more specific for the patient and to generate possible progressions of symptoms (such as mapping cancer growth and spread) in order to plan alternatives and decide on the best course of intervention.

This research could then be taken further by continuing to train the system as an imaging

modality alternative; starting with the base system that will have learnt to map a 3D model to an endoscopic image, the system could then learn to map to alternative modalities such as CTs and MRIs (as opposed to CT to MRI directly [24]). By building this translation from a 3D model to an image of variable modality the 3D model could then be manipulated and therefore, generate images from any required angle as opposed to the traditional axial, coronal and sagittal planes. Additionally, this system could be used as an alternative to CT and ultrasounds for areas such as the lung where they may not work, for example, to estimate lung usage.

As mentioned in Chapter 2, Generative Adversarial Networks (GANs) [15] have been successful in generating near realistic data, which then may be used to supplement model training [16]. In particular, Zhu et al. [19] proposed CycleGan, an unsupervised method to generate image-to-image translation requiring only unpaired images to generate realistic "looking" mappings between domains; photo to painting (e.g. Monet) translation is one such successful application of CycleGan that is analogous to this research. In this case, the endoscopic images form the photo domain and the rendered model the painting domain; the 3D model was manually created in Blender [96] where a virtual camera was used to set up the viewport before each rendering was taken. These rendered "slices" were taken at various locations within the colon model and although the rendered images appear to be realistic, they are still an "artistic representation" and is not the same as per those actual scans. However, the generated endoscopic images must be structurally accurate too, which plain CycleGan cannot achieve as there are multiple potential mappings that reduce the loss, this leads to ambiguity in the true, structurally congruent, translation and is therefore unusable in practical applications, such as in the case of CT to MRI translation. However, Chen et al. [24] combined CycleGan with One-shot learning, thereby (only) requiring a single pair, tackling the issue of ambiguity in image synthesis; unfortunately, this work cannot utilise One-shot learning due to the infeasibility of obtaining a pair without using a scanner, thereby making the system redundant. Despite these issues, the results from such systems evince promise for domain translation based on CycleGAN. Finally, reinforcement learning has been used to condition the input to a GAN for point cloud completion [22], an analogous task to our work in the form of semi-real to realistic translation, showing promise in both speed and accuracy.

Therefore, CycleGAN was chosen as the base architecture for this system; CycleGAN requires only images that belong to each domain and does not require paired images or even data from the same patient; this allows for far easier dataset collection. The CycleGAN system consists of a pair of GANs trained to map their input domain to the input domain of the other. These GANs are further regulated by additional loss functions on their ability to be used by one another, this coincidentally is the goal of the overall system. In this system, domain A comprises 224 real endoscopic colon images from a subset of the "CVC-ClinicDB" dataset [97], split 7:3 (train:test) with domain B consisting of 64 renders, from different angles, of the model, split 3:1.

Note that for the rest of this section "domain A" will refer to the real images of the colon and "domain B" refers to the renders of the model.

3.3.1 Model To Image Generation Methods

Since CycleGAN consists of two GANs with additional loss functions, it was natural to use the same U-Net architecture used in the segmentation network (section 3.2.2), for the generators. However, for the discriminators, the same encoder architecture was utilised, although the layers were trained from scratch as opposed to using the pre-trained MobileNetV2 layers used for the generators [2], followed by a final convolutional layer, with a leakyReLU activation layer, and ended by a single neuron to decide whether the output was real or generated (fig 3.16).

Additionally, in order to aid with training a robust model, random jitter was applied to the images in both domains by resizing the images to a slightly larger size and then randomly cropping the image back to the correct size and finally, with random chance, mirroring horizontally.

3.3.2 Results

Figure 3.17 shows the initial results using the methods described above (section 3.3.1); there is a lot of black in the images caused by the black padding from Domain A images (the real

[2]This may well change in future refinement of the model

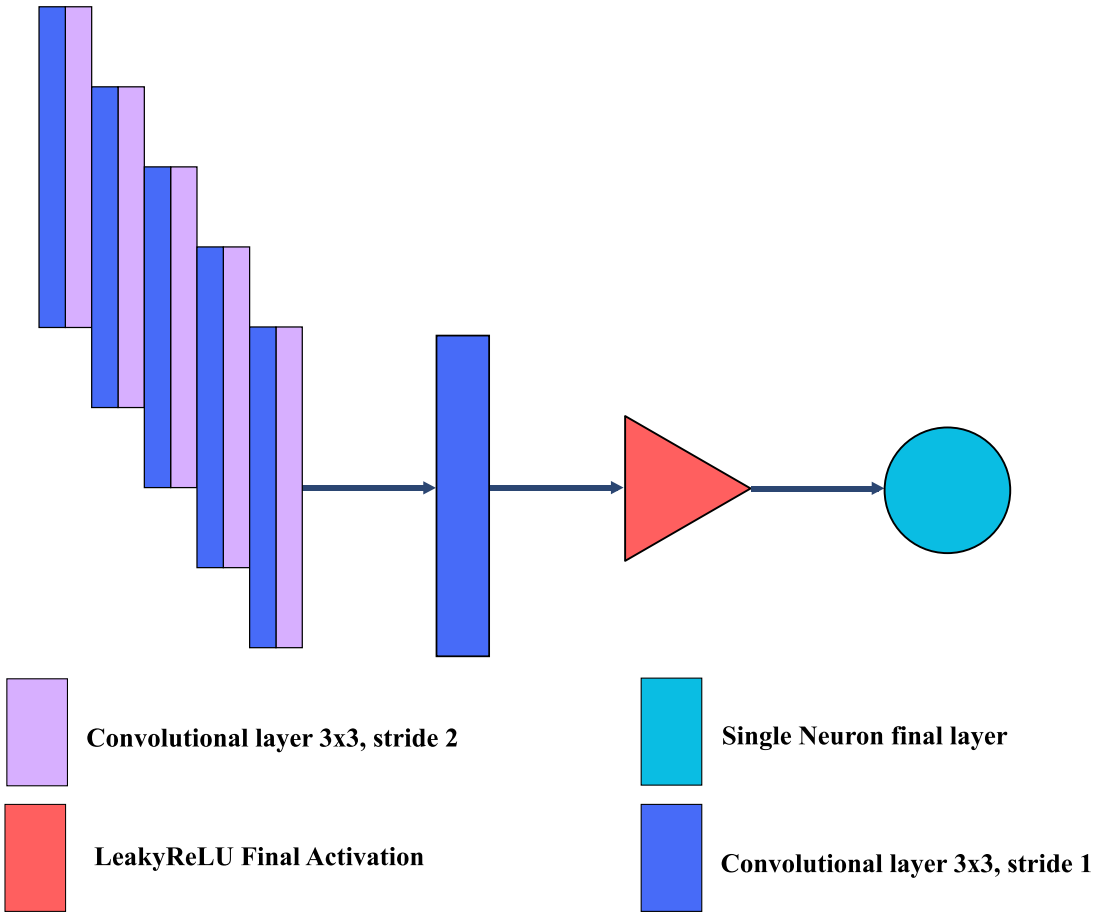
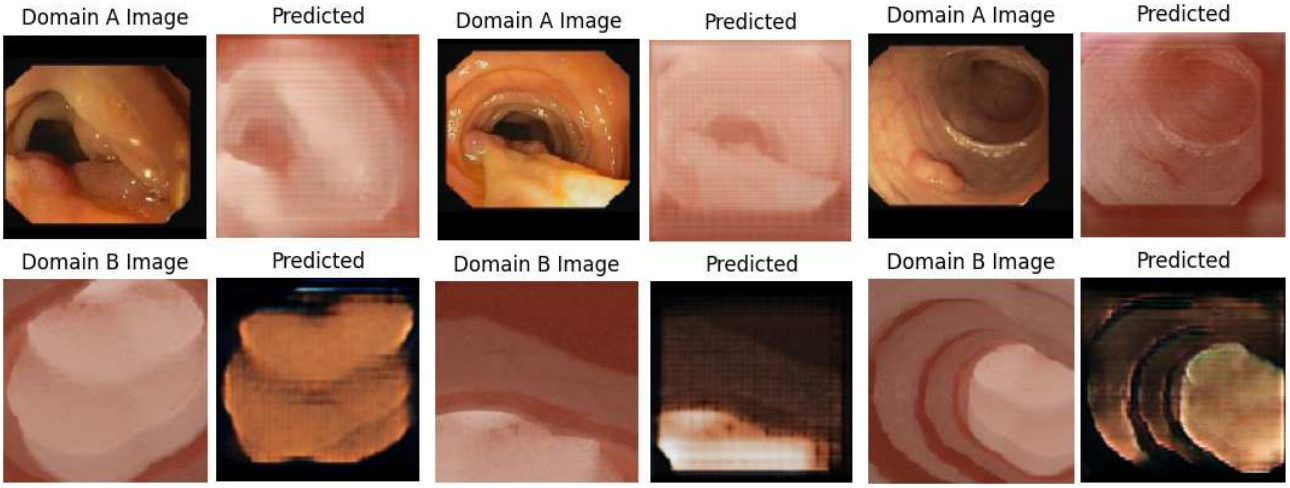


Figure 3.16: The Discriminator Network Architecture

images of the colon) formed in the resizing process as the original images had black corners from the endoscope's lens. Each result image is composed of four images, on the left is Domain A (top) and Domain B (bottom) and on the right is the output of CycleGAN (i.e. image as model (top) and model as image (bottom)). Whilst the top right image is unused as an output, it is essential in training and as such, is shown for comparison and is somewhat interesting to see how the network works; it appears to be attempting to recolour Domain A as opposed to shifting any geometry or relighting.

It is clear from these results that transfer learning results in a better network (fig 3.17c) as there are some specular highlights beginning to show through on the artificial Domain A image (the output obtained from a Domain B input). The next step was to replace the black padding,



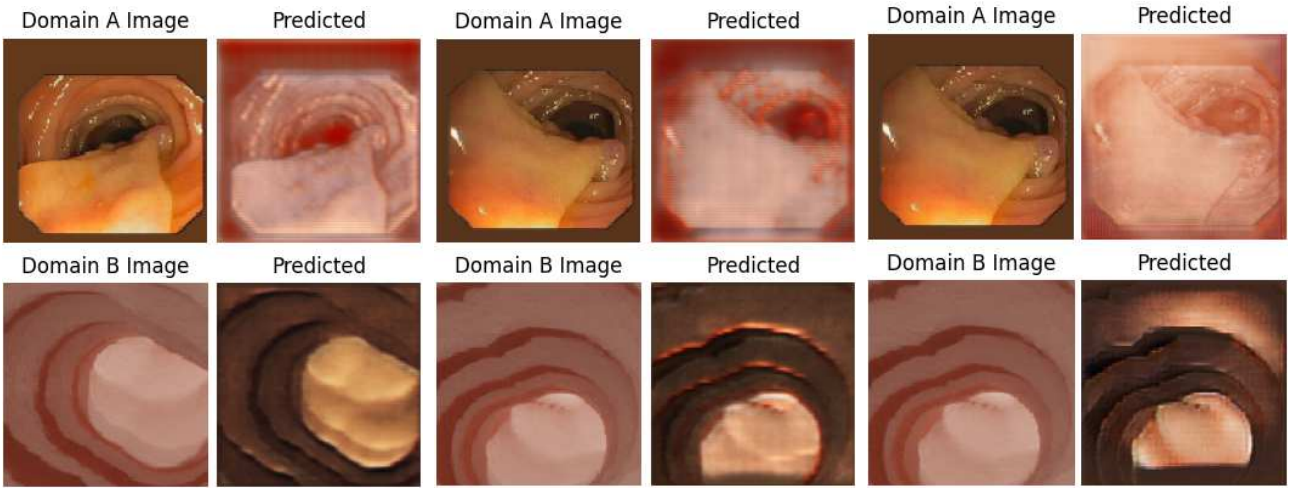
(a) Network 1 (b) Network 2 (c) Network 3 Transfer Learning

Figure 3.17: Results of CycleGAN for the first three networks. Networks 1 and 2 were trained from scratch, with the difference between the two being the final encoder layer, with Network 1 using 4 * 4 * 1280 (as with the segmentation layer) whereas Network 2 used 4 * 4 * 320 instead. On the other hand Network 3 used transfer learning from the same network used in the segmentation network with the same final encoding layer size as Network 2 (i.e. 4 * 4 * 320) to improve the output. None of the Networks had a bias term in the encoder and all networks were trained for 100 epochs each.

used in the Domain A images, with the mean colour in the image. This may well be refined in later versions, even potentially requiring an additional network; however, as it stands, the results were far better by using this colour change. The final network (fig 3.18b and 3.18c) generated far more realistic specular highlights, fig 3.18c shows the results from more training iterations on the final model and has the best results.

3.3.3 Conclusion

While this research demonstrates significant promise, several challenges must be addressed before the complete system can be fully realized. One such challenge lies in the simplicity of the current model, which utilizes a basic CycleGAN architecture. Although the initial results are encouraging, further augmentation is necessary to produce more realistic and clinically relevant images. For instance, integrating autoencoders into the system could accelerate the generator's performance [22] and allow for more precise tuning of the input-output processes. Additionally, as previously noted, applying transfer learning to the discriminators, similar to



(a) Network 4 after 100 Epochs (b) Network 5 after 40 epochs (c) Network 5 after 150 epochs

Figure 3.18: Results of the final two CycleGAN networks on images with non-black backgrounds. Note that both models used transfer learning in the same way network 3 did in the previous figure (3.17). Network 4 was identical to Network 3 except it was trained with the non-black background images. Network 5 on the other hand did use the bias term in its encoder which resulted in accurate specular highlights.

the approach that yielded substantial improvements in the generators, could further enhance the system's performance. Furthermore, incorporating another network to refine features such as the border colour of endoscopic images may also improve image realism and utility.

A more fundamental challenge, however, lies in the scarcity of raw data. The generation of artificial training data is inherently constrained by the availability of original training samples. In the medical domain, this limitation is exacerbated by stringent privacy protections and the disproportionate representation of positive versus negative cases in datasets. Therefore, any successful solution must effectively address these issues, ensuring data accessibility and diversity without compromising privacy or ethical standards.

3.4 Discussion

Considering the challenges uncovered by the research in this chapter, it is evident that a successful solution necessitates collaboration among multiple hospitals. Individual hospitals often have access to limited and imbalanced datasets, as anastomotic leaks are rare events that ideally should not occur, and when they do, their diagnosis is challenging, making labelled datasets

difficult to obtain. The key, therefore, lies in securely combining the data resources of multiple institutions. This requires a system that ensures privacy while enabling participants to pool their collective knowledge securely.

The logical approach is for each hospital to independently train its own network, thereby avoiding direct data sharing while leveraging the increased computational resources contributed by all participants. These locally trained networks can then be aggregated to form a global consensus model, benefiting from diverse and distributed data. This aligns precisely with the principles of federated learning, and this chapter has demonstrated that the natural progression of this research must be toward federated learning as a solution. Additionally, considering the ubiquity and utility of IoT devices, coupled with the advantages of edge computing, this approach enables real-time intraoperative training and classification directly at the edge. Federated learning emerges as the foundational paradigm required to make such a system both viable and effective and while federated learning does not inherently utilise distributed computing, this can be integrated seamlessly and is discussed in the following chapters.

In the following chapter, the integration of federated learning within hospital environments is discussed in detail, with an emphasis on a novel decentralised paradigm designed to address these challenges comprehensively.

Chapter 4

IoT Federated Blockchain Learning at the Edge

The technical contributions of this chapter are:

- A decentralised federated learning framework, for learning on the edge ensuring data privacy.
- A novel IoT federated learning system tested on the CIFAR-10 dataset using a physical device to obtain practical, and not simulated, results.

Please refer to section 4.1.1 for a full description of the contributions this chapter has made to the field. The following papers were published based on the research in this chapter:

1. **Calo James**, and Lo Benny. "Federated Blockchain Learning at the Edge" [**J**] In Information 14, no. 6: 318. 2023. doi: 10.3390/info14060318
2. **Calo James**, and Lo Benny. "IoT Federated Blockchain Learning at the Edge" [**C**] In Proceedings of IEEE EMBC 2023, 24-28 July 2023, Sydney Australia.
   doi: 10.1109/embc40787.2023.10339946

4.1 Introduction

As discussed in the previous chapter, the ability to effectively diagnose an anastomotic leak, a challenging task even for trained medical professionals, relies on fostering multi-institutional collaboration and leveraging the untapped potential of underutilised IoT devices in healthcare environments.

IoT devices are increasingly prevalent [98] and hold significant potential in the medical field, particularly within machine learning applications, yet they remain underutilised. These low-cost, energy-efficient, small, and intelligent devices [99] offer unrivalled benefits, with the healthcare industry projected to save up to $300 billion by leveraging IoMT (Internet of Medical Things) devices, particularly in managing chronic illnesses [100]. The market for IoMT is growing rapidly, from $28 billion in 2017 to an anticipated $135 billion by 2025 [100], driven by the increasing capacity of IoT devices to handle complex tasks once limited to full-sized computers. This rapid expansion underscores the importance of a solution that effectively harnesses this infrastructure in a secure, distributed, and intelligent manner to maximise the impact of IoMT in healthcare.

Machine learning techniques, particularly artificial neural networks (NNs), require large volumes of training data to achieve optimal performance; however, privacy concerns often limit the availability of such data, leading to suboptimal models or even insufficient data for training viable systems. This challenge is especially pronounced in healthcare, where patient data is heavily restricted and requires an ethics comity to determine what is appropriate to share. As a result, datasets are often homogeneous, drawn from small patient groups or localised to single hospitals. While neural networks are increasingly adopted in practice, including within the IoT paradigm, training these models at the edge on IoT devices remains challenging due to computational constraints and the need for substantial data. Distributed machine learning at the edge presents a compelling solution, particularly in healthcare, where the industry trend aligns with leveraging edge computing. However, existing systems often fail to integrate these capabilities effectively, instead addressing distinct aspects in isolation.

Therefore, a privacy-ensuring, low-powered and generalisable method to train neural networks whilst in use is required: federated learning. Federated learning is an effective method for training networks, without the training data ever being stored, which allows multiple networks to train on unique data and aggregate their weights to obtain a global network with similar accuracy as a network trained the traditional way. The downside of vanilla federated learning is the requirement on a centralised server for the aggregation; when privacy is involved, trust can cause complications. Therefore, a decentralised federated learning framework is realisable by combining federated learning with the decentralised nature of blockchain. This framework can run efficiently on multiple IoT devices across a P2P network removing the need for a trusted centralised server and enabling the learning process to be distributed across participating devices.

This approach has the potential to revolutionise paradigms within surgical practices by enabling individual neural networks to be trained securely, as data remains localised and is never stored or transmitted externally. By aggregating these locally trained models into a global network, the resulting model performs as if it has been trained on the complete dataset, an otherwise impractical scenario, while benefiting from increased diversity in the training data exposed to the global network. Additionally, distributing the training process, a capability that federated learning does not inherently provide, mitigates the computational limitations of IoT devices by leveraging their parallel operation. This strategy allows for the utilisation of underused idle devices, such as administrative desktops commonly found in hospital environments, thereby enhancing efficiency and resource usage.

This chapter focuses on a novel distributed federated learning framework for IoT devices, specifically within the Internet of Medical Things (IoMT), that combines federated learning and blockchain to enable decentralised, privacy-preserving, and efficient edge learning. This method, termed Learning on the Edge (LotE), addresses the challenge of training machine learning systems, particularly neural networks, directly on IoT devices by leveraging their ubiquity and parallel capabilities. Federated learning allows multiple networks with the same architecture to train simultaneously on non-iid (non-independent and identically distributed) data while aggregating their knowledge to produce a global model, effectively learning at the edge.

By integrating blockchain, this framework moves away from traditional centralised federated learning, replacing the centralised server with a decentralised scheme that enhances privacy, improves data integrity, and provides robustness. Furthermore, by transitioning from cloud-based architectures to edge computing, this framework supports secure and efficient training on sensitive patient data while utilising idle devices in a hospital environment, thereby addressing computational limitations and ensuring scalable, secure, and privacy-focused machine learning in healthcare.

The framework is designed for three paradigms:

1. Training neural networks on IoT devices to allow for collaborative training of a shared model whilst decoupling the learning from the dataset [101] to ensure privacy [102]. Training is performed in an online manner simultaneously amongst all participants, allowing for training of actual data that may not have been present in a dataset collected in the traditional way and dynamically adapt the system whilst it is being trained.

2. Training of an IoMT system in a fully private manner, such as to mitigate the issue with confidentiality of medical data and to build robust, and potentially bespoke [61], networks where not much, if any, data exists.

3. Distribution of the actual network's training, something federated learning itself does not do, to allow hospitals, for example, to utilise their spare computing resources for training.

4.1.1 Contributions

In response to the issues previously discussed in Chapter 3 this framework requires three architectural layers to achieve effective federated machine learning at the edge. The first layer, the federated layer, enables edge devices to communicate and train collaboratively. While federated learning is not inherently distributed, it is well-suited to distributed processing. In single-unit nodes, such as individual hospitals where data is collectively owned by the institution and privacy concerns are minimised, a centralised server can effectively support federated learning.

However, when multiple nodes, such as several hospitals participating in a study, are involved, privacy becomes critical, and establishing trust among participants is challenging. Federated learning mitigates the need for direct data sharing but still requires a trust-free system for model aggregation. This is addressed by the blockchain layer, which introduces a decentralised paradigm for federated aggregation, eliminating the need for trust among participants. By combining the federated and blockchain layers, this approach enables a robust and decentralised system that is less prone to overfitting and outperforms traditional centralised (non-federated) training methods on more computationally capable edge devices, such as laptops and desktops.

While this is a significant advancement in addressing key challenges in medical machine learning, such as data scarcity and privacy concerns, the third layer, the IoT mist layer, is critical for extending these capabilities. The IoT mist layer facilitates training directly on the IoT devices that capture the data, eliminating the need for persistent data storage. It enables a consortium of IoT devices to collaboratively train neural networks and allows individual devices to train multiple networks, ensuring scalability and security. Together, these three layers form an integrated system optimised for federated learning at the edge in healthcare.

Therefore, the main contributions of this chapter are:

1. A federated learning framework, for learning on the edge (LotE), that is fully decentralised, leveraging the blockchain layer, ensuring that the data is private, secured against malicious attacks, needs no trust between participants and requires only a small percentage of edge devices to be active at any one time; this enables devices that run infrequently or on a schedule to still participate without hindering the training process. This framework can be used with machine learning models, either directly via the framework or through the C API, developed simultaneously, allowing users to use other popular machine learning frameworks, such as TensorFlow.

2. A novel IoT federated learning system, using Tensorflow Lite on top of the aforementioned framework, to develop a configurable application for the training of neural networks on IoT devices and tested on the CIFAR-10 dataset [103] using a physical Pixel 4 Android smartphone running Android 13 with a Qualcomm Snapdragon 855 Octa-core CPU (1

x 2.84 GHz Kryo 485 Gold Prime & 3 x 2.42 GHz Kryo 485 Gold & 4 x 1.78 GHz Kryo 485 Silver) to obtain practical, and not simulated, results. As this application uses Tensorflow Lite any existing models built with Tensorflow can be converted to a Tensorflow Lite model and will therefore be fully compatible with the application with next to no overhead allowing for further research.

4.2 Methods

4.2.1 Federated learning

In the conventional setting, the objective of a neural network is to approximate a target function by minimising the prediction loss with respect to the network's parameters.

$$\min_{\omega \in \mathbb{R}^d} \mathcal{F}(\omega) = \ell(\boldsymbol{x}, \boldsymbol{y}, \omega) \tag{4.1}$$

Where ℓ is a chosen loss function, consistent across all participating networks, $\boldsymbol{x}, \boldsymbol{y}$ are the training (input and desired output) vectors and ω is a list of the network's weights; as all participants have the same network architecture ω has a fixed length.

However, with federated learning, there are many participating networks training (potentially simultaneously) to form a global network; this global network has not seen any training data, ensuring data privacy, and is the result of aggregating the participating local networks. This aggregation requires all participants to upload their network's weights alongside the number of training examples they have been trained on. This may be collated on a centralised server or, as is the case in this research, added to a block to be mined on the blockchain (as described in section 4.2.2). These participating networks are then aggregated using the *FedAvg* algorithm [14] (equation 4.2) to obtain the weights for the global network; this new network may then be disseminated to each participant and used for the next round of federated learning.

$$\omega_{\text{global}} \triangleq \frac{1}{\sum_{i=1}^{N} |\chi_i|} \cdot \left(\sum_{i=1}^{N} |\chi_i| \cdot \omega_i \right) \tag{4.2}$$

Where for N participating networks, $|\chi_i|$ is the cardinality of the set of training examples (i.e. the number of examples seen) by network i (for $i \in \{1..N\}$) and ω_i is the list of weights of network i, trained as defined in equation 4.1. Since only the number of examples seen by any participant ($|\chi_i|$) is added to a block and not the dataset itself (χ_i), no data is ever shared. Note that $\sum_{i=1}^{N} |\chi_i| \equiv |\bigcup_{i=1}^{N} \chi_i|$.

4.2.2 Decentralisation with blockchain

One of the greatest weaknesses of vanilla federated learning is the requirement on a centralised server; to address this the blockchain layer tweaks the federated paradigm to a decentralised distributed server architecture. Blockchain is a decentralized, distributed ledger technology that serves as the foundation for many cryptocurrencies, including Bitcoin. While the terms blockchain and Bitcoin are often used interchangeably, they refer to different concepts. Bitcoin is a cryptocurrency, an application that operates on a blockchain to facilitate secure, decentralized transactions. In contrast, blockchain is the underlying technology that enables Bitcoin and other applications by providing a tamper-resistant, transparent method of recording and verifying data. A useful (though simplified) analogy is to consider blockchain as the transport layer and Bitcoin as an application running on top of it, similar to how the OSI model separates network communication layers. Bitcoin itself is highlighted as it was the first and most well-known implementation of blockchain technology, demonstrating its viability for secure, decentralized systems. Additionally, its blockchain protocol settings were adapted for the framework developed in this chapter as discussed in the following subsections. Consequentially, this enables shifting the logical architecture from the cloud/fog, which essentially comprises of devices connected to a server, to the edge, where every device is independent and autonomous; the system will continue to work with only one node and even if all nodes go down, the system can recover fully, since each node contains a copy of the accepted blockchain, this may not

be possible if the central server lost its data. The following details this layer's architecture regarding the fundamental components of the blockchain.

4.2.2.1 Block

The block's format closely mirrors Bitcoin's blockchain protocol, which is robust and concise, with two major changes: The target formula and the federated components. The target is used to decide when the block has been mined via proof of work (PoW), which was chosen over an alternative, more green (both environmentally and chronologically) consensus mechanisms, such as proof of stake (PoS) [104, 105, 106]. Unfortunately, the downside of PoW is the energy cost, a miner who performs PoW must continually guess, in a deterministic manner, a hash that is less than the target, since the target is in big endian hexadecimal format we refer to it as a hash with more leading zeros than the target.

The criticism stems from the high computational cost that provides no actual benefit, other than to make it infeasible for a malicious node to pervert the system. However, in an IoT system, this is not as problematic; with hundreds of mining (IoT) devices, the problem can be split across them, much like mining pools. Additionally, PoW provides a high level of security and, unlike PoS, cannot be easily manipulated by individuals, the computational cost of PoW provides an additional deterrent when combined with the federated layer: the more resources put towards mining the less there are for the local training rounds encouraging sharing of the mining responsibilities.

Moreover, since the blockchain is being utilised as a trust mechanism for federated learning, the mining target difficulty can remain lower, reducing the computational cost; this additionally increases the rate at which blocks are added to the chain which in turn reduces the time between each aggregation step. Therefore, lower-powered devices will have enough computing resources to generate hashes, while still providing the same level of protection. As a result, the application uses a block rate of approximately one every 1.5 minutes. This rate is long enough for multiple local updates, from different sources, to be included in a block, prior to the block being added to the chain, without being so long that either the global network becomes outdated or a local

device, that misses the aggregation round, grows stale.

PoS, on the other hand, would not be as suitable since it relies too heavily on transactions, doesn't include a mining step, would give too much power to larger institutions, whose networks are likely to be less diverse, and promotes coin hoarding which negates the bonus benefit of blockchain, rewards: This is what incentivizes hospitals to utilise their spare computing power.

4.2.2.2 Mining

Mining the local updates via PoW requires storing the target in the block; however, the true target size has the same number of bits as the hash. Therefore, much like Bitcoin, the target is encoded in 4 bytes:

$$\boldsymbol{Target} \overset{hex}{=} 0\mathrm{x}\overbrace{\phi_1\phi_2}^{\Phi}\overbrace{\theta_1\theta_2\theta_3\theta_4\theta_5\theta_6}^{\Theta} \triangleq \Theta * 2^{8*(\Phi-4)} \tag{4.3}$$

Where the first byte ($\phi_1\phi_2$) is an exponential scale and the lower three bytes contain the linear scale. As with Bitcoin, we scale the exponential by 8, since there are 8 bits in a byte, which simplifies the bit manipulation calculations. However, unlike Bitcoin which scales the exponential by 3, the application scales the exponent instead by 4 to generate target values at the lower end of the spectrum due to using a lower block mining rate than Bitcoin.

4.2.2.3 Cryptography

Blockchain is built upon the fundamental concept of cryptographic hashes, a mathematical function that maps an input of arbitrary size to a fixed-size output, called the hash value, in a deterministic manner that is collision-resistant and irreversible (one-way) $h : \{0,1\}^* \rightarrow \{0,1\}^n$ (where $\{0,1\}^*$ is the set of all possible binary strings of arbitrary length and $\{0,1\}^n$ is the fixed size output binary string of length n). For example, SHA-256 produces a 256-bit (32-byte) hash, taking in any arbitrary length binary string and outputs exactly 256 bits. The output of

the hash function should not be similar in any way to any other output formed from a similar input.

This framework uses the Keccak-256 cryptographic hash instead of SHA256 and RIPEMD160, which Bitcoin uses. Keccak-256 is stronger compared to both and is used by Ethereum, which is a distributed state machine as opposed to a distributed ledger. Since further research into adopting some of Ethereum's changes to the plain distributed ledger, such as smart contracts, would be highly beneficial for this field of study, keccak-256 is a more natural fit. However, double hashing is still used. Hashing (applying a hash function in order to generate an output hash) is required in order to secure the chain by making it easy to verify that the block hashes to the specified value but being complex enough that a malicious entity cannot produce the correct hash if they modified the block.

4.2.2.4 Networking

The peer-to-peer (P2P) network used for communication provides an essential benefit; whilst a single node will still be functional, the benefits of federated learning would be severely reduced. P2P networks are ideally suited to handle two vital situations: First, each node must be able to request a copy of the blockchain, including a list of addresses of other nodes, which will receive the address of the connecting node and, secondly, the ability to broadcast information to all nodes accessing the chain, even those that may not be directly connected to this node.

By using a pair of UDP sockets (algorithm 1), the communication can be parallelised and enables communication to be distributed amongst different (local) devices. A hospital, for example, may have many IoMT devices but none with networking capabilities, just Bluetooth; they could therefore connect all IoMT devices to a single, network capable, IoT device which would handle the networking and packet forwarding, much like Network Address Translation (NAT) with regards to WiFi routers. Consequently, any IoT device may participate, with the only requirement being a connection to a networking node, potentially via Bluetooth or even hardwired to a communication module, at some point downstream. Moreover, if a collection of IoT devices train as one unit, only one device needs to connect to the outbound UDP connection

with all devices gaining the benefits. The blockchain must be accessible to all nodes but does not need to be persistently stored on every individual node. Each participant maintains a copy of the chain that can be shared among their own nodes and transmitted to external nodes via the P2P network. The chain can be stored on any suitable storage medium, such as a shared drive within a hospital's IT network, provided that internal nodes can retrieve and forward it upon request. This approach ensures accessibility while preventing IoT devices from having to store large amounts of data.

Algorithm 1 UDP Pair Communication

```
procedure INBOUND
    loop
        msg ← incomingMsg
        if ValidateIsNewestChain(msg) then
            chain ← msg
        else if msg ∉ addresses then
            addresses.append(msg)
            Outbound(Inbound.Address, msg)
        end if
    end loop
end procedure

procedure OUTBOUND(msg, addr = NULL)
    if msg == JoinNetwork then
        Broadcast(Join + Inbound.Address)
    else if addr == NULL then
        Broadcast(msg)
    else
        SendDirectMsg(msg, addr)
    end if
end procedure
```

4.2.3 IoT Federated learning

All devices contain an instance of a, potentially pre-trained, TensorFlow Lite model and train on incoming data, private to each participant. After a number of epochs, each participant may submit their network's weights and the number of different examples seen to the blockchain (figure 4.1). The chain then contains the globally aggregated model that can then be used as the new initial network for the next round of local training.

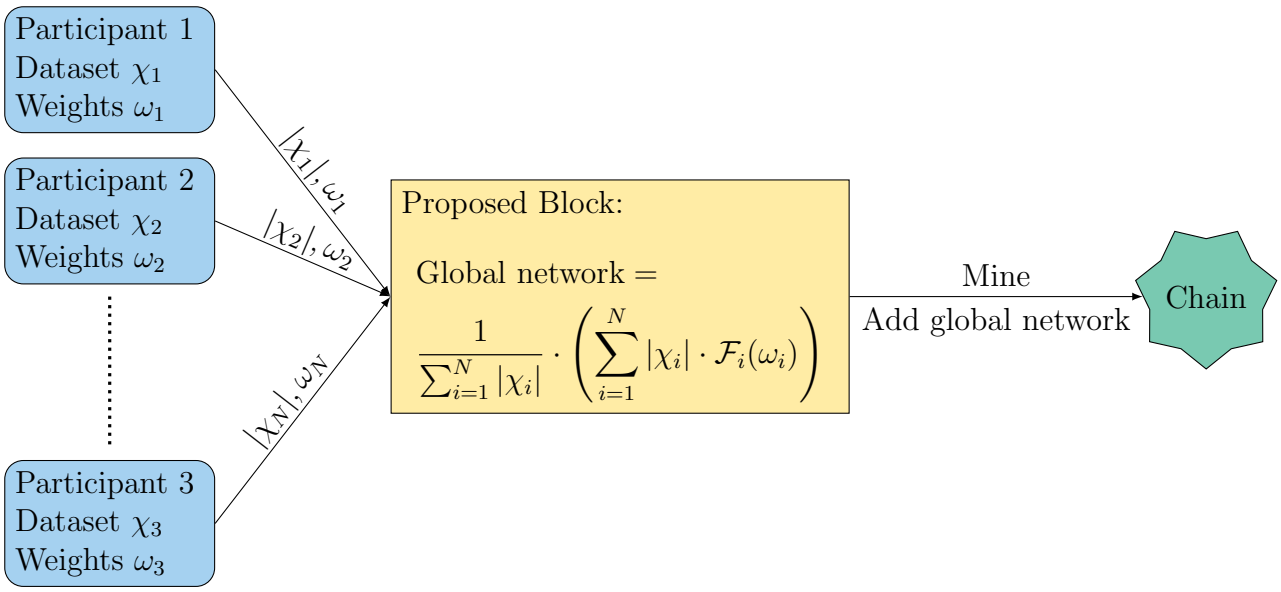


Figure 4.1: Clients contributing to the blockchain using (4.1). Note $\bigcup_{i=1}^{N} \chi_i \subseteq$ all possible data.

4.3 IoT System Complexity

This section analyses the complexity and latency of the proposed IoT system including actual timings taken from the system running in debug mode on a physical Pixel 4 Android smartphone running Android 13 with a Qualcomm Snapdragon 855 Octa-core CPU (1 x 2.84 GHz Kryo 485 Gold Prime & 3 x 2.42 GHz Kryo 485 Gold & 4 x 1.78 GHz Kryo 485 Silver) training multiple networks to perform image classification on the CIFAR-10 dataset [103] in order to give real, practical values to fairly evaluate the system.

The first step of the system is the local training; using γ to denote the number of CPU cycles to execute one step of training (i.e. applying backpropagation to one input) then the complexity of an epoch of local training is defined as

$$\Theta_{\text{local training}} = \max_{\mathcal{D}_i \in \mathcal{D}} \left(\frac{\beta_{\mathcal{D}_i} \cdot \gamma}{\sigma \mathcal{D}_i} \right) \tag{4.4}$$

where $\beta_{\mathcal{D}_i}$ is the batch size of the input data used in each epoch, $\mathcal{D}_i$ is a participating device (from the set of all devices $\mathcal{D}$) and $\sigma \mathcal{D}_i$ is the CPU frequency of the participating device $\mathcal{D}_k$. In our experiments, (averaging over 144 epochs) the latency of $\Theta_{\text{local training}} = 23.68$ seconds.

Each participant then sends their model to the blockchain for validation which consists of the PoW consensus algorithm followed by the aggregation of all models into a global model (using equation 4.2). Therefore, the complexity of the global aggregation is as follows:

$$\Theta_{\text{global aggregation}} = \frac{\rho + \gamma_{agg}}{\sigma_{\mathcal{V}}} \tag{4.5}$$

where ρ is the number of CPU cycles to execute the PoW algorithm, $\gamma_{agg} = (N+1)\cdot\Xi + (N-1)\cdot\lambda$ is the number of CPU cycles to execute the aggregation of N models (where Ξ and λ denote the number of CPU cycles to execute to perform scalar matrix multiplication and matrix addition respectively) and $\sigma\mathcal{V}$ is the CPU frequency of the validator. In our experiments, consisting of four models, the latency of $\Theta_{global\_aggregation} = 106.83$ milliseconds which is a negligible cost owing to the large target for the PoW algorithm. However, in general, $\rho >> \omega_{agg}$ unless there are a large number of participating devices.

Running the entire system on our single device (sequentially, with no concurrency or optimisations applied) yielded a complete run time of 19.79 minutes consisting of four models training for 50 epochs each with global aggregation occurring every 25 epochs (i.e. all participants were aggregated twice in the complete run).

4.4 Results

To test the base federated learning framework, TensorFlow was used to design a simple and small network architecture, comprising two convolutional and max pooling layers, a final convolutional layer and two dense layers, to classify the CIFAR-10 and PneumoniaMnist datasets. Using both standard and federated training paradigms, the same network architecture was used in five experiments, per dataset, using 10%, 25%, 50%, 75% and 100% of the training data; in the federated case, the data was shared equally amongst all participating models, such that no two models saw the same data points, as would be the case in a live system (especially when using image data as the input). For each subset of the data, the model was trained for 150

epochs whilst additionally measuring the effect of altering the number of epochs each participating member trained for before the federated update and the number of participants to the federated scheme as shown in Table 4.1 and 4.2. The resulting accuracies in each sub-table are all within a small range showing that federated learning produces similar results to the standard method but with the benefit of being applicable to distribution and working on different (albeit similar) datasets with no shared data points. Furthermore, when training using federated updates, with each "'local training"' round being performed sequentially, the training process runs quicker compared with the standard method due to each participant operating on smaller subsets of the data, allowing for optimisations such as better caching; this is especially apparent on smaller devices.

Following these results, the same network architecture was used for training directly on an IoT device (the same phone used in section 4.3 to evaluate the system complexity). Ordinarily, each participating network would be first pre-trained by TensorFlow before being converted to the TensorFlow Lite format. However, the application should be trained almost entirely on-device and so only one epoch (against 100% of the CIFAR-10 training data) to have a starting point over a network with completely randomised weights; however, pre-training of another instance of the network with 100 epochs was also obtained to compare the effects of pre-training. The results of non-federated learning on these two networks are shown in Table 4.3 with the loss and accuracy after training further on the device (against only 25% of CIFAR-10's training data) for 50, 100 and 150 epochs. These results imply that the pre-training caused the model to overfit and is likely only useful if the data the device observes will differ from the pre-training data.

Next, multiple instances of the single epoch pre-trained network were trained via federated learning. Globally updating all participating networks after either 25 or 50 epochs and training for 50, 100 and 150 epochs, the federated setting for 2, 4 and 8 participating networks was tested. All participants were trained on an even split of 25% of CIFAR-10's training data such that no two networks saw the same example (an example of the case with 8 participating networks is shown in figure 4.2. The results in Table 4.4 show that the accuracy is very similar to the non-federated context with the main difference within each configuration being that the

Update	Number of Models	Accuracy
25	2	50.57%
25	4	51.04%
25	8	51.33%
50	2	50.37%
50	4	51.3%
50	8	52.33%
75	2	50.95%
75	4	49.03%
75	8	52.90%
Non-Federated		52.04%

(a) Trained with 10% of the training data

Update	Number of Models	Accuracy
25	2	57.69%
25	4	57.01%
25	8	57.87%
50	2	57.81%
50	4	57.95%
50	8	58.16%
75	2	58.54%
75	4	59.51%
75	8	59.43%
Non-Federated		56.94%

(b) Trained with 25% of the training data

Update	Number of Models	Accuracy
25	2	63.16%
25	4	64.29%
25	8	63.53%
50	2	63.06%
50	4	64.24%
50	8	62.33%
75	2	63.93%
75	4	63.64%
75	8	63.89%
Non-Federated		62.62%

(c) Trained with 50% of the training data

Update	Number of Models	Accuracy
25	2	65.91%
25	4	66.06%
25	8	65.83%
50	2	66.46%
50	4	65.65%
50	8	65.76%
75	2	65.61%
75	4	66.86%
75	8	65.51%
Non-Federated		65.88%

(d) Trained with 75% of the training data

Update	Number of Models	Accuracy
25	2	68.09%
25	4	68.75%
25	8	68.85%
50	2	69.23%
50	4	68.12%
50	8	68.11%
75	2	68.56%
75	4	68.09%
75	8	68.07%
Non-Federated		67.64%

(e) Trained with 100% of the training data

Table 4.1: Accuracy of convolutional model on CIFAR-10 after 150 epochs. The top accuracy in each case is highlighted.

Update	Models	Accuracy	Recall	Precision
25	2	62.50%	100%	62.50%
25	4	62.50%	100%	62.50%
25	8	62.50%	100%	62.50%
50	2	62.50%	100%	62.50%
50	4	62.50%	100%	62.50%
50	8	62.50%	100%	62.50%
75	2	78.37%	97.44%	75.25%
75	4	62.50%	100%	62.50%
75	8	62.50%	100%	62.50%
Non-Federated		83.81%	95.90%	81.48%

(a) Trained with 10% of the training data

Update	Models	Accuracy	Recall	Precision
25	2	81.89%	97.69%	78.56%
25	4	62.50%	100%	62.50%
25	8	62.50%	100%	62.50%
50	2	81.57%	98.72%	77.78%
50	4	71.79%	99.23%	69.11%
50	8	62.50%	100%	62.50%
75	2	80.93%	98.72%	77.15%
75	4	72.12%	98.97%	69.42%
75	8	62.50%	100%	62.50%
Non-Federated		81.89%	98.72%	78.09%

(b) Trained with 25% of the training data

Update	Models	Accuracy	Recall	Precision
25	2	83.97%	97.44%	80.85%
25	4	86.06%	95.38%	84.35%
25	8	62.50%	100%	62.50%
50	2	83.97%	97.69%	80.72%
50	4	84.78%	95.90%	82.56%
50	8	83.97%	94.87%	82.22%
75	2	84.46%	97.18%	81.51%
75	4	84.62%	97.18%	81.96%
75	8	67.63%	100%	65.88%
Non-Federated		86.70%	97.69%	83.74%

(c) Trained with 50% of the training data

Update	Models	Accuracy	Recall	Precision
25	2	84.62%	97.44%	81.55%
25	4	84.29%	97.44%	81.20%
25	8	86.54%	94.87%	85.25%
50	2	85.10%	97.69%	81.94%
50	4	83.65%	96.67%	80.90%
50	8	85.74%	95.90%	83.67%
75	2	84.29%	97.44%	81.20%
75	4	84.62%	96.92%	81.82%
75	8	79.17%	99.23%	75.29%
Non-Federated		85.10%	97.95%	81.80%

(d) Trained with 75% of the training data

Update	Number of Models	Accuracy	Recall	Precision
25	2	86.22%	96.92%	83.63%
25	4	86.06%	98.21%	82.72%
25	8	86.54%	95.90%	84.62%
50	2	86.22%	96.92%	83.63%
50	4	84.78%	97.69%	81.58%
50	8	86.54%	95.64%	84.77%
75	2	85.58%	96.92%	82.89%
75	4	86.22%	96.41%	83.93%
75	8	86.38%	95.64%	84.58%
Non-Federated		86.38%	97.95%	83.22%

(e) Trained with 100% of the training data

Table 4.2: Accuracy of a small convolutional model (92,737 parameters), with the same architecture as the previous network (table 4.1), trained on PneumoniaMnist after 150 epochs. The top accuracy in each case is highlighted. Unlike the CIFAR-10 results the federated case does not always surpass the non-federated case due to the size of the dataset (CIFAR-10 has 50,000 training images as opposed to PneumoniaMnist's 5,232 training images. However, even with an extremely constrained model, federation does not adversely affect the results (given enough data) and is therefore a viable method when a single institution does not have access to enough data.

Pre-trained epochs	On-device trained epochs	Final Loss	Final Accuracy
	50	1.43	48.04%
1	100	1.65	49.43%
	150	3.08	47.97%
	50	1.73	48.02%
100	100	3.43	46.94%
	150	4.42	46.67%

Table 4.3: Loss and accuracy of a neural network against CIFAR-10 test data trained and evaluated on an Android phone. Each network was pre-trained on a laptop for the specific number of epochs on 100% of the CIFAR-10 training dataset and then trained further on the Android phone for the specified number of epochs against 25% of the CIFAR-10 training data. While it may appear that this model is randomly guessing, this is not the case as there are 10 classes. Please refer to table 4.4 for the mathematical validation of these models.

loss increases with more training despite the accuracy remaining virtually constant. This is due to the gradients becoming either too small (vanishing gradients) or too large (exploding gradients), making weight updates ineffective and unstable. This is a result of the way IoT devices handle floating-point operations; these are often less precise compared to dedicated computers, using fewer bits, as memory is at a premium and rounding affects the training process. While these results are not close to the state-of-the-art developed by Yang et al. [63], which obtained over 70% accuracy on CIFAR-10, the network used to generate these results is significantly smaller and is running on a physical device, which could not possibly store the AlexNet network, containing over 60 million parameters, that was used in the state-of-the-art which itself was not tested on a physical IoT device but instead was run on a cluster computing system at the fog layer.

4.5 Discussion

The findings in this chapter demonstrate that artificial neural networks trained using the proposed framework outperform those trained with traditional, centralised (non-federated) methods. These networks are also less prone to overfitting, owing to the federated aggregation process. This establishes that the framework is not only a viable alternative to traditional training paradigms but also a significant improvement, offering substantial benefits within ecosystems, such as hospitals. For example, in a clinical situation, where privacy, security, and the handling

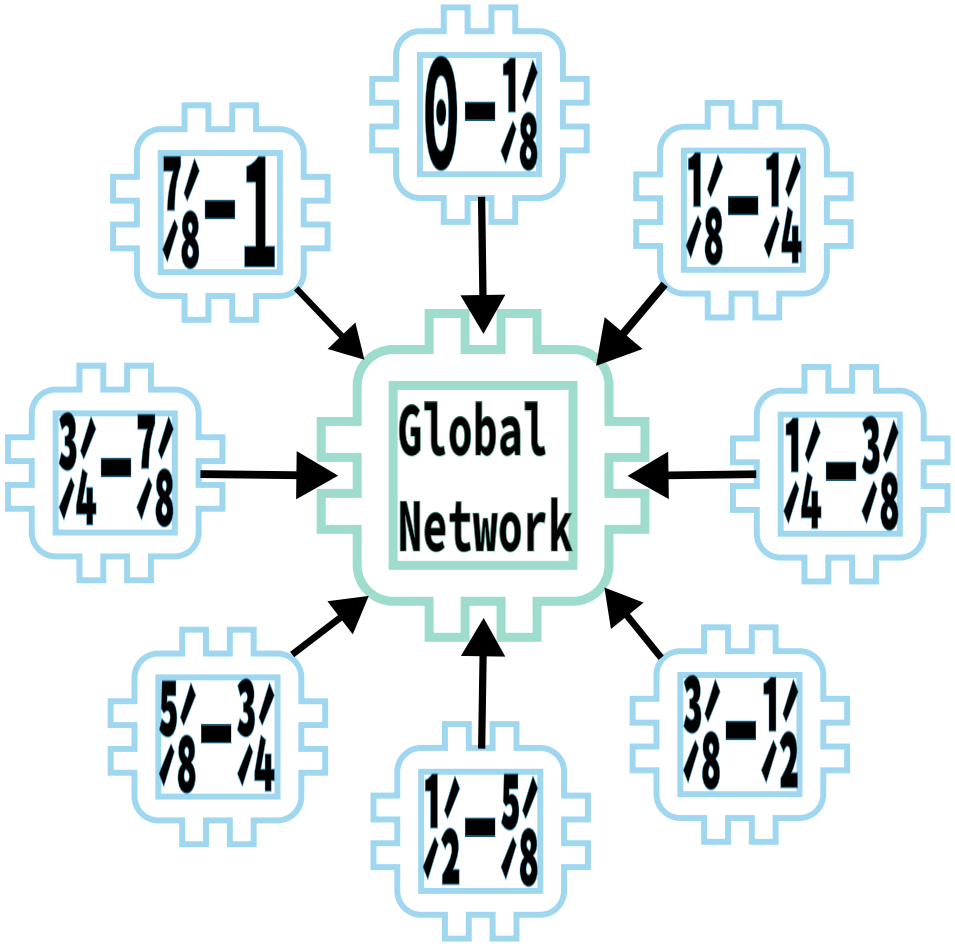


Figure 4.2: Federated Training of 8 models with an even split of the dataset. Given a dataset, χ, each local network, $i \in \{1..N\}$, trains on an even split of the dataset proportional to the number of networks such that for $\chi = \{\chi_1, \chi_2, .., \chi_k\}$, network i is trained on $\{\chi_{(i-1)\cdot k}, .., \chi_{i\cdot k}\}$. These networks are then aggregated into a global network that performs as if it had been trained on χ despite not receiving any data and thereby preserving privacy.

of heterogeneous datasets, often limited in size, are critical challenges, the framework provides a robust and effective solution.

Additionally, the framework facilitated the development of a fully customisable Android application that enables on-device training of arbitrary neural networks using TensorFlow Lite. This innovation allows existing networks to be seamlessly integrated into the federated learning framework and utilised securely on Android devices, enhancing privacy and security. Despite the networks was trained on the IoT device (in this case, an Android phone) outperforming their non-federated counterparts, their classification performance, based on training on an even

split of only 25% of the full dataset, remains limited compared to the results achieved on a laptop. The laptop, representing the upper limit of edge computing capabilities in this context, underscores the need for additional strategies to bridge the computational power gap and enhance the performance of IoT devices in federated learning scenarios.

This research has highlighted multiple intriguing avenues for future research, for instance, exploring federated learning's capability to handle truly homogeneous datasets: rather than splitting a dataset like CIFAR-10 into random subsets of equal length, a potential alternative would involve assigning each of the 10 participating networks a single class to train on exclusively. This approach could provide valuable insights into the robustness and adaptability of federated learning in scenarios where data distributions are entirely non-overlapping.

However, it is evident that enabling the sharing of spare computing resources for distributing "local training" is paramount; it is therefore essential to secure the training data effectively. This can be achieved either by operating on encrypted data, for example, through homomorphic encryption, or by transforming the data into a non-reversible representation, such as using Fourier or wavelet transformations. These methods would facilitate the integration of blockchain technologies, including smart contracts, to automate tasks and manage the sharing of processing capabilities. This approach could also incentivize participation by offering additional utility to collaborators.

In the subsequent chapter, the adaptation of masked autoencoders [107] is explored as a means of obfuscating data, enabling more sophisticated federated aggregation while fully leveraging the blockchain layer. This innovation allows the training process to be distributed not only across internal IoT devices but also to other nodes, including those from external hospitals, without compromising the privacy guarantees of federated learning. Critically, no reconstructable form of the training data ever leaves the originating device. This system enables hospitals to utilise spare computational power, for instance from administrative PCs, and even trade these resources with one another, fostering greater collaboration and advancing shared objectives in a secure and efficient manner.

Epochs per Global Update	Participating Networks	Total Epochs	Loss	Accuracy
25	2	50	1.43	48.04%
		100	1.66	49.44%
		150	3.07	48.20%
50	2	50	1.43	48.03%
		100	1.66	49.19%
		150	3.05	47.93%
25	4	50	1.43	48.04%
		100	1.64	49.62%
		150	3.06	48.12%
50	4	50	1.43	48.04%
		100	1.64	49.65%
		150	3.08	48.10%
25	8	50	1.43	48.04%
		100	1.65	49.46%
		150	3.08	48.28%
50	8	50	1.43	48.18%
		100	1.65	49.33%
		150	3.03	47.97%

Table 4.4: Loss and accuracy of a neural network against CIFAR-10 test data trained via federation and evaluated on an Android phone. Each participating network was trained on and even split of 25% of the CIFAR-10 training dataset with no participant seeing the same data.While it may appear that these models are randomly guessing, there are 10 outcome classes and therefore these models are is significantly better than random. Due to the large sample size (CIFAR-10 has a test size of 10,000 images) the binomial distribution can be approximated with a normal distribution allowing testing to determine if the model is better than random by using a z-test for proportions. The Z-score measures how many standard deviations the observed result is from the expected mean under a null hypothesis. In this case, the null hypothesis assumes that the classifier is randomly guessing (i.e., achieving only 10% accuracy). The Z-score formula for proportions is: $Z = \frac{\hat{p}-p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$ where $\hat{p} = 0.4965$ (the observed accuracy of the top model), $p_0 = 0.1$ (the null hypothesis / the expected random accuracy) and $n = 10000$ (the number of samples / test images). Therefore, as $Z = \frac{0.4965-0.1}{\sqrt{\frac{0.1*0.9}{10000}}} = \frac{0.3965}{0.003} \approx 137.17$, the observed accuracy of 49.65% is 132 standard deviations above the outcome if the classifier were guessing randomly. In a normal distribution, results beyond Z = 1.96 are considered statistically significant at the 95% confidence level. A Z-score of 132.17 is astronomically high, making it virtually impossible that these results occurred by random guessing.

Chapter 5

Proof of Reasoning (PoR) for Privacy Enhanced Federated Blockchain Learning at the Edge

The technical contributions of this chapter are:

- PoR, a novel consensus mechanism that mitigates the effect of malicious networks and utilizes useful computation, as opposed to PoW.
- Evidence that masked autoencoders can continue to use a high masking proportion in downstream tasks without catastrophic effect on the accuracy of linear-probing.
- Combining masked autoencoders and PoR, more complex aggregation methods can be used, as shown in section 5.2.3.2.

Please refer to section 5.1.1 for a full description of the contributions this chapter has made to the field. The following paper, based on the research in this chapter, has been submitted to IEEE Internet of Things Journal and is currently awaiting a response:

1. **Calo James**, and Lo Benny. "Proof of Reasoning for Privacy Enhanced Federated

Blockchain Learning at the Edge" [**J**] In IEEE Internet of Things Journal (IEEE IoTJ) (Proposed, not yet accpeted)

5.1 Introduction

Building on the discussion in the previous chapter, this chapter addresses the critical need to leverage spare computing resources effectively for secure distributed training. Ensuring privacy during such resource sharing requires robust obfuscation techniques to prevent data reconstruction without compromising its utility. Traditional encryption methods, while secure, often impose constraints on the architecture of neural networks, limiting their flexibility and applicability. In contrast, this chapter introduces a novel application of masked autoencoders (MAEs) to encode raw data. This approach not only renders reconstruction infeasible but also generates enriched feature maps that enhance the performance of downstream classifiers, creating a balance between privacy preservation and computational efficiency.

Consensus mechanisms are the core of any blockchain system. However, the majority of these mechanisms do not target federated learning directly nor do they aid in the aggregation step. This chapter introduces Proof of Reasoning (PoR), a novel consensus mechanism specifically designed for federated learning using blockchain, aimed at preserving data privacy, defending against malicious attacks, and enhancing the validation of participating networks. Unlike generic blockchain consensus mechanisms commonly found in the literature, PoR integrates three distinct processes tailored for federated learning. Firstly, a masked autoencoder (MAE) is trained to generate an encoder that functions as a feature map and obfuscates input data, rendering it resistant to human reconstruction and model inversion attacks. Secondly, a streamlined downstream classifier is trained at the edge, receiving input from the trained encoder. The downstream network's weights, a single encoded datapoint, the network's output on this input, and the ground truth are then added to a block for federated aggregation. Lastly, this data facilitates the aggregation of all participating networks, enabling more complex and verifiable aggregation methods than previously possible. This three-stage process results in more robust

networks with significantly reduced computational complexity, maintaining high accuracy by training only the downstream classifier at the edge.

The fundamental concept of blockchain is the consensus mechanism, how does a collective decide what is the truth. Interestingly, this is closely related to the issue of spotting weak or even malicious networks that would corrupt the federated aggregation. While it is possible to combine consensus mechanisms with algorithms such as multi-KRUM[63] to reduce the effect malicious networks have on the final aggregated network they do so by elimination. Unfortunately, the distinction between malicious networks and non-conforming networks is impossible; obviously, it is beneficial to remove malicious networks but non-conforming networks may have useful knowledge. Unfortunately, they may appear malicious as their dataset could be quite different from the majority, a consequence of using non-iid (non-independent and identically distributed) datasets. Additionally, these methods still require a consensus mechanism such as Proof of Work (PoW) or Proof of Stake (PoS) that are not designed with federated learning in mind.

The trade-off for decentralizing federated learning by utilizing blockchain is transparency; every block in the chain is publicly available, making privacy a significant challenge. Validating a participating network requires domain-specific input, which may be difficult to obtain or unavailable for sharing. This is especially true in the medical field, where sharing raw images is unacceptable. One approach to mitigate this issue is to encode the input data; however, it is crucial that this data cannot be decoded, whether by brute force or a model inversion attack. Techniques such as homomorphic encryption excel in this regard but necessitate increased computational complexity to enable the matrix multiplication required for neural networks and is incompatible with non-linear activation functions such as rectified linear units (ReLU).

To ensure the privacy of the data while enabling federated learning, masked autoencoders (MAE) [107] are utilised to generate a feature map and continue to mask a high percentage of the input [31] in contrast to classical MAE, which does not mask the input outside of training, resulting in an encoded subarray of patches from the original image. This method provides high semantic content to the downstream classifier, maintains accuracy, and is highly resistant

to attempts to reconstruct the original, unencoded data.

5.1.1 Contributions

This chapter focuses on a privacy-enhancing, resilient paradigm, enabling efficient utilization of IoT resources to support machine learning training at the edge for healthcare/medical applications resulting in a novel framework, integrating a consensus mechanism designed solely for federated learning, named Proof of Reasoning (PoR) by utilising masked autoencoders (MAE) [107] to allow privacy-aware data sharing and minimize computational complexity. The main contributions of this new framework are:

- PoR, a novel consensus mechanism that mitigates the effect of malicious networks and utilizes useful computation, as opposed to PoW, while resulting in similar guarantees of the infeasibility of replacing the entire chain with false data.
- Evidence that masked autoencoders can continue to use a high masking proportion in downstream tasks without catastrophic effect on the accuracy of linear-probing and as a result allow the use of the masked and encoded data for validation and aggregation of the downstream classifier.
- Combining masked autoencoders and PoR, more complex aggregation methods can be used to combine the participating (local) networks into a global (aggregated) network (an example of which is shown in section as shown in section 5.2.3.2).

5.1.2 System Overview

The following sections in this chapter delve into the system's operation in detail; however, Figure 5.1 provides an illustrative example of how the final federated blockchain learning system operates with three participants, each employing distinct methods for training their local networks.

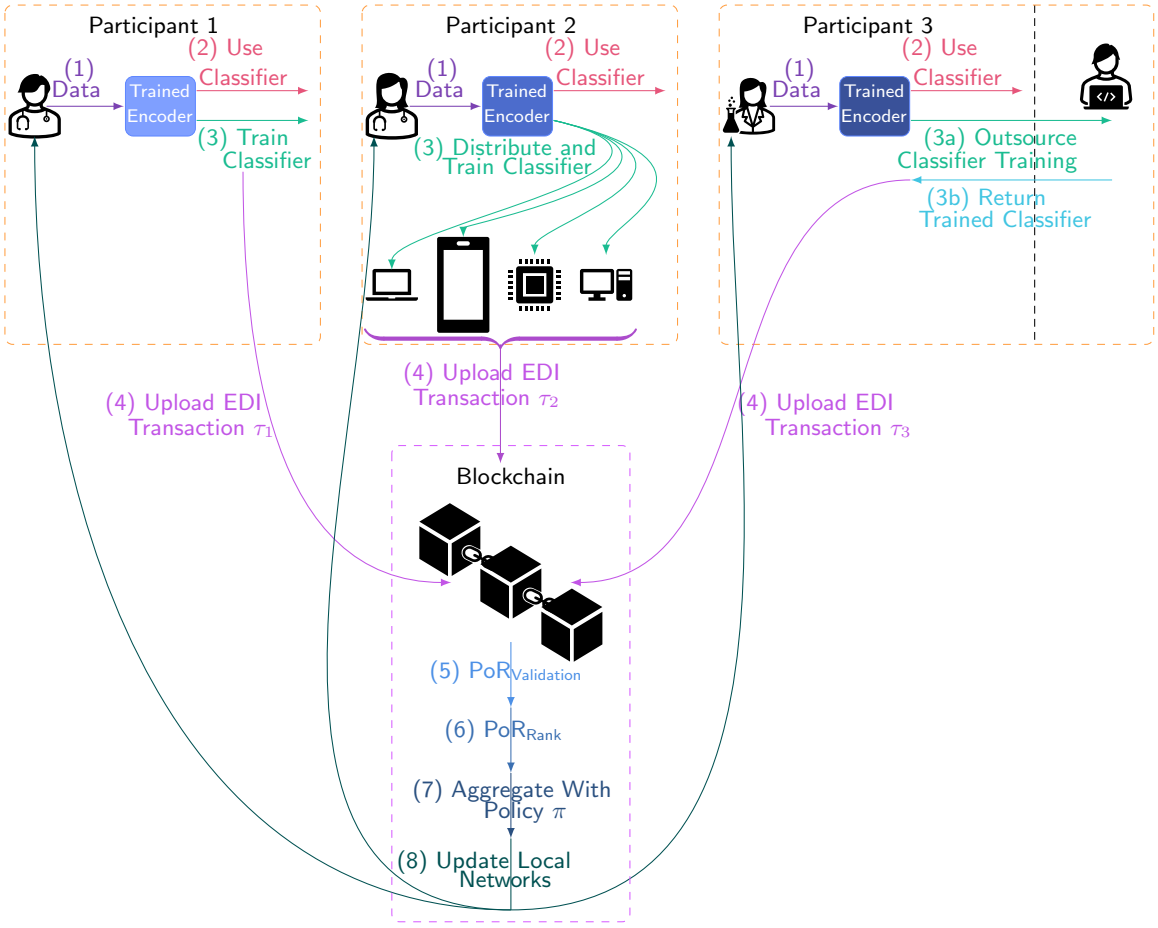


Figure 5.1: An example of PoR being used by three participants.

1. **Data Encoding:** Each participant encodes their data, potentially collected intraoperatively, using their individually trained encoder transformer (Step 1). The encoded data is then passed to their downstream classifier for immediate use (Step 2), while simultaneously serving as input for the next round of local training (Step 3).

2. **Local Training Variations:**

   - *Participant 1* conducts training entirely on their local infrastructure, leveraging computational resources directly within their institution.
   - *Participant 2* utilizes a distributed IoT mist network to train their classifier. Devices in the network may employ standard distributed training algorithms, such as the All-Reduce algorithm, or engage in local federated learning. Alternatively, each IoT device can act as an independent participant as each processes only a subset of the encoded data.

- *Participant 3* outsources the training of their downstream classifier to another institution. They share only the encoded data and their local model, ensuring privacy while leveraging external computational resources.

3. **Blockchain Integration:** After completing local training, each participant uploads an Encoder-Decoder Interface (EDI) transaction to the blockchain (Step 4). The EDI transaction includes all necessary information as detailed in Section 5.2.3.1. If Participant 2 treats each IoT device as an independent participant, multiple EDI transactions, one per device, are submitted to the blockchain.

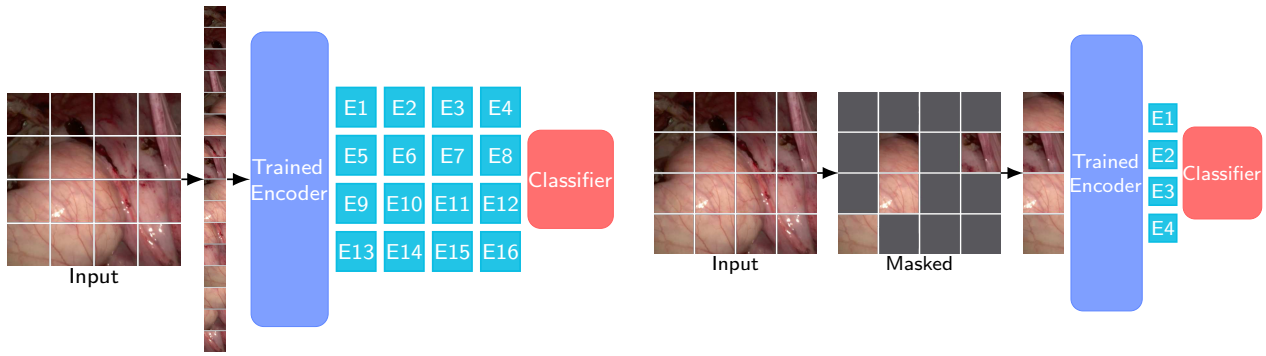
4. **Consensus and Validation:** The blockchain employs the Proof of Reasoning (PoR) consensus mechanism to validate each uploaded network against an encoded example datapoint provided by the participant (Step 5). This step ensures the integrity of contributions while maintaining privacy.

5. **Ranking and Aggregation:** The validated networks are ranked (Step 6) based on their performance across all uploaded datapoints, following the specified aggregation policy π. The ranked networks are then aggregated into a global model (Step 7), which is distributed back to each participant, initiating the next round of local training.

This decentralized, privacy-preserving system enables robust collaboration across multiple institutions and IoT devices, optimizing resource utilization and enhancing model performance while safeguarding sensitive medical data.

5.2 Materials and Methods

Proof of Reasoning (PoR) is a consensus mechanism tailored to enable more advanced aggregation techniques with built-in validation for each participant. To facilitate privacy-enhanced, secure, and efficient processing at the edge, masked autoencoders (MAE) [107] are integrated into the framework. MAE generates a feature map, which is subsequently fed into a simplified



(a) Vanilla MAE (b) PoR MAE

Figure 5.2: MAE Training and upstream feature map usage. In contrast to vanilla MAE, where the input is split into unmasked patches which are then fed sequentially into a trained encoder resulting in an encoding per patch, PoR MAE continues to mask the input, as it was when training the encoder, and only the remaining (unmasked) patches are fed to the trained encoder resulting in significantly fewer encoded patches being passed to the classifier.

downstream classifier that participates in federated learning and training at the edge without the original encoder. This approach significantly reduces computational complexity during training and inference while maintaining competitive accuracy, as the feature map generated by the encoder retains essential information. For federated aggregation, the network's weights, a single encoded datapoint, the network's output for that datapoint, and the corresponding ground truth are encapsulated in a block to ensure seamless and secure integration.

Unlike traditional MAE implementations (Figure 5.2a), this design does not encode the entire image. Instead, it processes only the encodings of unmasked patches (Figure 5.2b). This approach inherently resists model inversion attacks, as the decoder introduces additional complexity during training (Figure 5.3). The reconstruction process depends on masked tokens and positional encodings, making it exceedingly difficult to infer the original unmasked patches from the encoded representation. The following sections provide a detailed description of this design and its contributions to enhancing federated learning in healthcare.

5.2.1 Pre-training

5.2.1.1 Input Augmentation

Following MAE [107] the training data is augmented by applying image resizing and random cropping to increase the variance in the data seen by the network, the input must be resized regardless to allow the correct number of patches to be extracted. Additional augmentations such as rotation or vertical flipping should not be applied as this can result in images that would never naturally occur adding additional uncertainty to the reconstruction used for self-supervised pre-training without resulting in any benefits. On the other hand, while horizontal flipping is a legitimate choice of augmentation, the masked reconstruction is an already complex task which may be negatively affected by additional input augmentation and was not used in the original paper, it was therefore decided to not include this or any additional input augmentation strategies.

5.2.1.2 Masking Strategy

Since the masked autoencoder is used as a feature map and data obfuscator the downstream classifier does not receive all the encoded data. Fortunately, the masking strategy used in training the masked autoencoder can differ from the strategy used to generate the feature map. For example, when training the masked autoencoder, random masking could be applied in order to get a different set of patches each time the same image is encountered during training. Once the encoder is trained, this masking strategy can be replaced with one that masks each patch proportionally, based on a chosen heuristic (for example one that ensures information dense patches are not masked) as the now trained encoder is used as a feature map generator feeding the encoded patches into the downstream classifier. Experiments were performed with both random masking and masking proportionally to the correlation of each patch to all other patches; however, in both training the MAE and the downstream classifier, random masking (with a normal distribution) provided superior results. With a given masking ratio μ_r, $\mu_r \cdot \rho$ patches were replaced with a learnable mask token.

Model Name	Heads	Width	Layers
ViT-Huge	16	1280	32
ViT-Large	16	1024	24
ViT-Base	12	768	12
Edi-Encoder	1	1280	2
Edi-Decoder	1	192	2

Table 5.1: Encoder and Decoder Transformer parameters vs standard ViT

5.2.1.3 Encoder-Decoder Design

The network architecture for both the encoder and decoder are based on Vision Transformers (ViT) [32]. However, typical ViTs, such as the standard variants ViT-Base, ViT-Large, and ViT-Huge, are computationally intensive for IoT applications, containing 86 million, 307 million, and 632 million parameters, respectively. To address this, the encoder has been significantly downsized to contain only 26.2 million parameters, approximately 30% of ViT-Base, by processing each patch individually. The decoder, on the other hand, while smaller in projection width, requires 85.5 million parameters as it must process both masked and unmasked patches simultaneously. Importantly, the decoder's larger size does not hinder deployment since it is used solely during pre-training and is not required once the encoder is trained. To strike a balance between computational feasibility and efficiency, the entire MAE network was trained on a laptop, reflecting the computational resources accessible in hospital environments (Table 5.1).

Following the patching and masking of the input image, as outlined in [107], each patch's position embedding is combined with its projection, consistent with standard ViT practices [32]. Specifically, a single dense layer projects the unmasked patch's pixels, while a learnable mask token is projected for masked patches (Figure 5.3). The unmasked patches are then collated and passed through sequentially to the encoder. The encoder's output is subsequently recombined with the masked tokens, their original order encoded within, and passed into the decoder for input reconstruction, ensuring the preservation of spatial relationships and enabling effective pre-training.

The chosen loss function applies the mean squared error function (MSE_ℓ) between the original

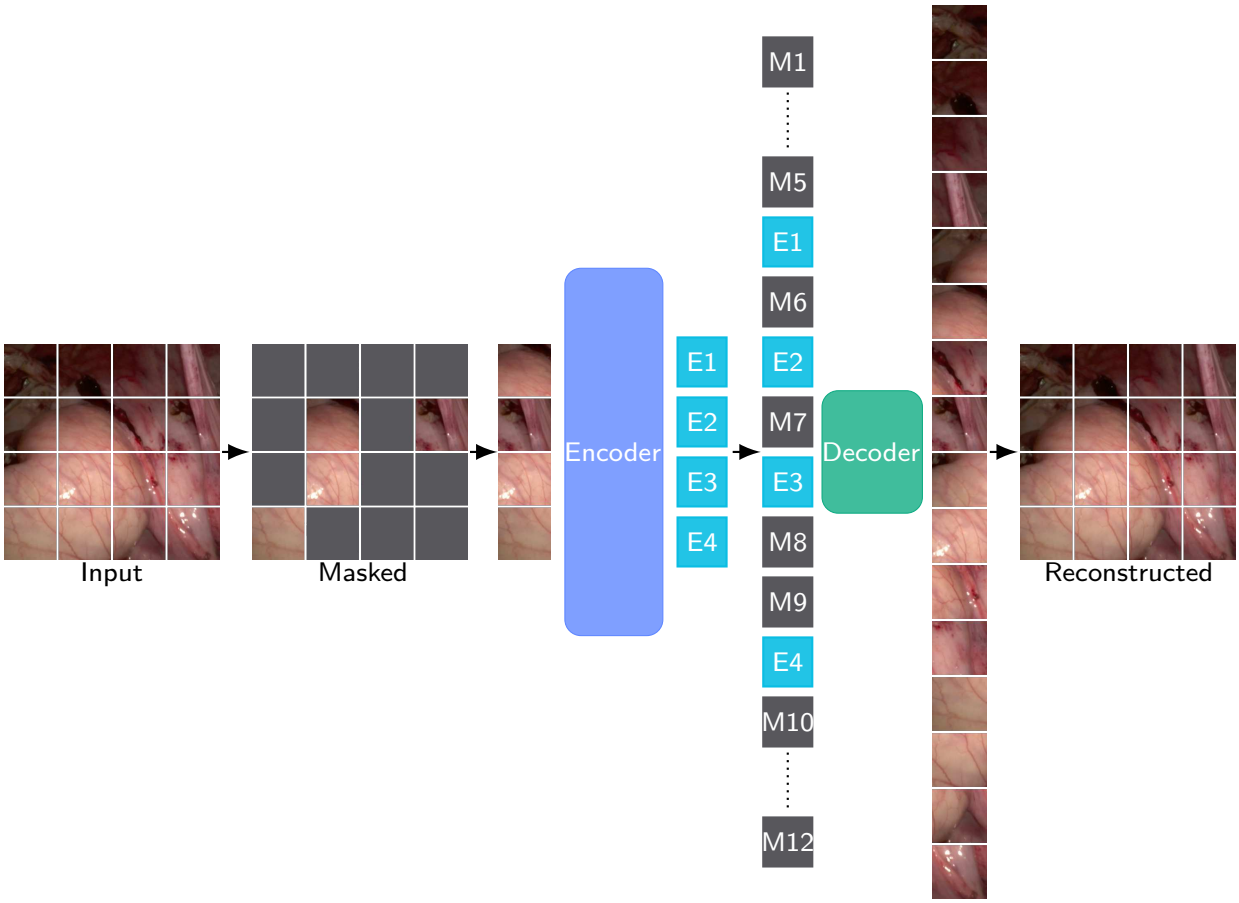


Figure 5.3: Self-supervised training process of a masked autoencoder (MAE): The input image is divided into patches, with a high percentage masked out. The unmasked patches are assigned positional encodings and processed by the encoder, which transforms them into encoded representations. These encoded patches are then combined with masked patches, each containing a learnable mask token and its positional encoding, and passed to the decoder, which reconstructs the original image. This training enables the encoder to extract meaningful representations from limited input data.

image and the reconstruction for the masked patches only; this is the same as MAE [107]; while mean absolute error (MAE_ℓ[1]) resulted in a higher reconstruction accuracy, it produced a less accurate classifier when used as a feature detector.

[1] X_ℓ is used for error functions in order to differentiate Masked Autoencoders (MAE) from Mean Absolute Error (MAE_ℓ)

5.2.2 Training

5.2.2.1 Downstream Classifier Design

The downstream classifier operates as the federated component within the system, requiring a lightweight design to minimize computational overhead. To achieve this, a streamlined feed-forward neural network architecture was utilized. However, for some classification tasks, once federated learning was implemented, this architecture was not complex enough to perform in line with the non-federated models (for the experiments in Section 5.3 each modification to this simplified downstream model is detailed). In contrast to the standard Masked Autoencoder (MAE), where the entire input is encoded and processed by the classifier, this framework maintains masking on the input (Figure 5.2). This approach not only reduces the networks's size and complexity by decreasing the number of patches but also enhances privacy preservation. As the federated step involves testing the downstream classifiers submitted to the blockchain, some data must be uploaded. By masking and then encoding the data the framework ensures it is obfuscated, rendering it unrecognizable to humans and resistant to model inversion attacks, while still serving as a meaningful input for the downstream classifier.

5.2.3 Federation

5.2.3.1 Federated Transaction

For each participant $i \in \mathcal{P}$, Proof of Reasoning (PoR) introduces an Encoder-Decoder Interface (EDI) transaction τ_i that is uploaded to the blockchain (Figure 5.4). Each EDI transaction encapsulates the weights of the participant's downstream classifier, ω_i, and a list of one or more tuples, where each tuple contains an encoded datapoint κ_i, the output of the downstream classifier for that datapoint $\hat{y}_i$, and the actual classification label y_i. Let Ω_i denote a downstream

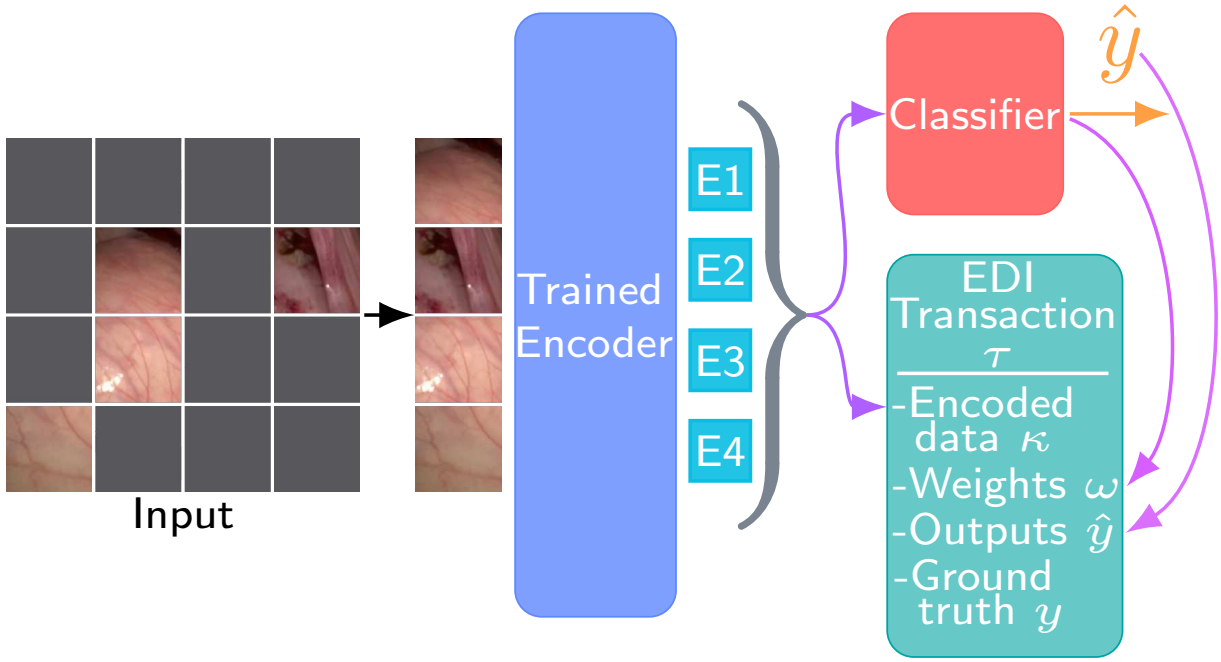


Figure 5.4: An example of generating an Encoder-Decoder Interface (EDI) transaction to be added to the blockchain using PoR. The unmasked patches of a single datapoint are encoded by the participant's trained encoder transformer and added to the transaction as the encoded data array κ. Additionally, the weights of the downstream classifier ω, the output of the downstream classifier on the encoded data $\hat{y}$ and the true classification of the input y are also added to the transaction.

classifier with weights ω_i then:

$$\tau_i = (\omega_i, [(\kappa_i, \hat{y}_i, y_i), \cdots]) \tag{5.1}$$

$$\hat{y}_i = \Omega_i(\kappa_i) \tag{5.2}$$

$$\kappa_i = \lambda_i(x_i) \tag{5.3}$$

Where λ_i represents a masked encoder and x_i is the original input. The EDI transaction does not include the masked encoder λ_i or the raw input x_i. Even with access to the transaction data $(\omega_i, [\kappa_i, \hat{y}_i, y_i), \cdots])$, it is computationally infeasible to derive x_i by inverting λ due to the one-way nature of the encoding process.

This structure enhances federated learning by enabling more sophisticated and verifiable aggregation methods. Unlike federated averaging (FedAvg) [14], which relies on unverifiable counts of training examples used by each participant, the EDI transaction provides transparency and

robustness, paving the way for novel aggregation schemes.

5.2.3.2 Proof of Reasoning

The PoR mechanism begins by verifying that each network in the participant pool $\mathcal{P}$ produces outputs within a specified tolerance ϵ, ensuring a baseline level of trustworthiness. While this step does not entirely eliminate the risk of malicious networks, since inputs could originate from a different domain, allowing potentially malicious networks to pass validation, the subsequent aggregation step is designed to minimize their impact. Each network is then scored and ranked according to an aggregation policy π, which operates on the set of all EDI transactions, $\mathcal{T}$, recorded on the blockchain. The aggregation policy π comprises two primary functions: a scoring function $\mathcal{S}^\pi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$, which evaluates the performance of each classifier Ω_i on all of the included encoded datapoints $\bigcup_{j\in\mathcal{P}}\{\kappa_j\}$, and a reduction function $\mathcal{R}^\pi : \mathbb{R}^n \to \mathbb{R}$, which consolidates these scores into a final ranking for each participant. Specifically, for each participant i:

$$\text{PoR}_{\text{Validation}} = |\Omega_i(\kappa_i) - \hat{y}_i| < \epsilon \tag{5.4}$$

$$\text{PoR}_{\text{Rank}} = \mathcal{R}^\pi(\{\mathcal{S}^\pi(\Omega_i(\kappa_j), y_j) | j \in \mathcal{P}\}) \tag{5.5}$$

A straightforward example of an aggregation policy for multiclass classification is a linear scoring method based on classification probabilities. In this approach, each network is evaluated by assigning points proportional to the placement of its predicted probability for the correct class. For instance, if a network's predicted probability for the correct class is the third highest among all predictions, it is awarded (num_classes - 2) points. The reduction function then aggregates these scores by summing them across all inputs to generate an overall ranking for each network. Algorithm 2 shows an example psudo-code implementation for the aforementioned linear scoring aggregation policy.

This method offers transparency and accountability, directly tying the performance of a model

Algorithm 2 Linear Scoring Aggregation Policy.
Line 2 sets the scores to range from *num_classes* (the maximum score) to 1 (the minimum score).
Line 3 sorts the classification output, (*prediction_logits_list*) which is the probability that the input belongs to the corresponding class number with that index, in descending order by index. i.e. the first element in the list $sorted\_classification$ is the class index that the network assigned the highest probability and the last element is the class index with the least probability.
Line 4 assigns the index that the expected class has in this sorted classification list. In the example where the network's predicted probability for the correct class is the third highest among all predictions the *score_index* would be 2.
Line 5 then returns the corresponding score from *score_weights* (in this case $num\_classes - 2$).

1: **procedure** SCORE($prediction\_logits\_list, expected\_class, num\_classes$)
2: $score\_weights \leftarrow [num\_classes, num\_classes - 1, .., 1]$
3: $sorted\_classification \leftarrow argsort(prediction\_logits\_list, DECENDING)$
4: $score\_index \leftarrow sorted\_classification.index(expected\_class)$
return $score\_weights[score\_index]$
5: **end procedure**

to its accuracy on encoded inputs. Unlike FedAvg, which relies on scaling a model's contribution based on the unverifiable count of examples it has processed, this scoring mechanism evaluates networks based on their actual output performance, ensuring a more equitable and verifiable aggregation process. By assigning each participant's downstream classifier a PoR Rank through this policy, the system supports more intelligent and robust aggregation strategies. These strategies address critical shortcomings of traditional methods like FedAvg, enhancing the reliability, fairness, and accountability of federated learning systems in complex and privacy-sensitive domains such as healthcare.

5.3 Results and Discussion

5.3.1 Cifar10 Experiments

As this framework is designed to operate within the IoT infrastructure of a hospital, the first requirement was to develop an encoder-decoder architecture optimized for a leaner computational environment, such as a laptop with decent but limited processing power compared to dedicated machine learning servers. This constraint necessitated reducing the number of heads and layers

Encoder			Decoder			Reconstruction Accuracy
Heads	Width	Layers	Heads	Width	Layers	
1	1024	2	1	192	2	15.95%
1	1024	2	1	208	2	15.99%
2	1024	4	2	128	4	15.62%
1	1280	2	1	192	2	16.01%
2	1280	2	1	128	2	14.80%
1	2048	2	1	128	2	15.30%

Table 5.2: Impact of varying the number of heads, width, and layers for both the encoder and decoder transformers: The encoder, being the core component used throughout the system, is designed to be larger than the decoder while maintaining efficiency to operate on resource-constrained IoT hardware. Both components must remain lightweight to ensure compatibility with edge devices. Each configuration was evaluated using a masking ratio of 75% on images from the Cifar-10 dataset.

in the Encoder-Decoder transformers. Table 5.2 illustrates the impact of various transformer parameters on the reconstruction accuracy of the Cifar-10 dataset [103], demonstrating the trade-offs involved in optimizing for constrained hardware.

Since PoR's implementation of the masked autoencoder (MAE) diverges from the original design in [107], extensive experiments were conducted to identify optimal transformer configurations and evaluate the influence of different training parameters. These experiments included variations in masking percentage, masking method, and loss function to assess their effects on downstream classifier performance. Results revealed that while higher masking percentages reduced reconstruction accuracy, they significantly improved downstream classifier accuracy. Consequently, a masking rate of 90% was chosen as the optimal balance. Moreover, using the MAE_ℓ loss function yielded higher reconstruction accuracy but detrimentally affected downstream classification performance. Masking methods based on correlation (e.g., selecting the most or least similar patches or even a 50/50 split of each) consistently degraded results, highlighting random masking as the superior approach despite its non-deterministic nature. Table 5.3 presents initial results for each parameter configuration, while Table 5.4 reports the reconstruction accuracy of the chosen MAE settings across various datasets.

For the downstream classifier, the initial design employed a simple feedforward network consisting of batch normalisation, global average pooling, one hidden layer, and an output layer. This architecture performed well in traditional (non-federated) machine learning settings and

Mask %	Masking Method	Loss Function	Loss	Reconstruction Accuracy
60%	Random	MSE_ℓ	0.0103	18.14%
60%	Random	$\mathbf{MAE_\ell}$	**0.0680**	**18.45%**
62.5%	Random	MSE_ℓ	0.0110	17.72%
75%	Random	MSE_ℓ	0.0142	16.08%
85%	Random	MSE_ℓ	0.0191	13.83%
85%	Correlation	MSE_ℓ	0.0259	9.87%
90%	Random	MSE_ℓ	0.0233	12.35%
95%	Random	MSE_ℓ	0.0307	10.32%

Table 5.3: Effect of masking percentage, masking method, and loss function on the reconstruction accuracy of the masked autoencoder (MAE): The experiments were conducted after training for 100 epochs on the Cifar-10 dataset (training set size: 40,000, testing set size: 10,000). While using MAE_ℓ led to higher reconstruction accuracy, employing MSE_ℓ with a high masking percentage of 90% resulted in superior classification accuracy for the downstream classifier. This indicates that, despite reduced reconstruction performance, the encoder learned a more effective representation for classification of the input data. This may seem counter-intuitive as the reconstruction accuracy is extremely low and the masking rate is very high; however, unlike the original MAE paper, the primary goal of this research is to maximise classification accuracy while maintaining a high masking ratio and therefore the loss and reconstruction accuracy were not directly used to determine parameter combinations and instead the affect of the resulting encoder on the downstream classification accuracy was used to decide the masking ratio and loss function for the encoder. This points to the fact that the reconstruction accuracy is not as important to classification tasks as a deeper semantic understanding of an image.

Dataset	No. Examples	MSE	MAE	Accuracy
Cifar-10	40,000	0.0233	0.1101	12.35%
Cifar-100	40,000	0.0238	0.1110	13.32%
ImageNet	60,000	0.0281	0.1245	8.63%
ChestMnist	80,000	0.0063	0.0525	51.58%

Table 5.4: Results for the reconstruction accuracy of the masked autoencoder after training for 100 epochs on various datasets with a masking percentage of 90%

proved effective when encoded datasets could be freely exchanged (Table 5.5). However, under a federated paradigm, the classifier's accuracy suffered due to limited data availability (e.g., when Cifar-10 was split among two or four participants) and the inherent variance in encoded datapoints. Despite incorporating a high dropout rate of 70%, which slightly improved results, the performance remained suboptimal (Table 5.6). Interestingly, if all participants used the same masked autoencoder pre-trained on the entire dataset, the system achieved results comparable to the non-federated case. However, this approach is impractical in federated learning due to privacy and data-sharing constraints, making transfer learning the logical next step to enhance system performance.

Hidden Layers	Hidden Width	Mask%	Accuracy
1	2560	50%	62.91%
1	4096	75%	59.90%
1	5120	50%	64.06%
1	5120	62.5%	61.00%
1	5120	75%	60.32%
1	8192	75%	59.24%
2	$\begin{pmatrix}5120\\256\end{pmatrix}$	75%	58.73%

(a) Effect of hidden layer configurations and input masking percentage on downstream classifier accuracy: The simplified downstream classifier (excluding the residual bottleneck layer) was evaluated with varying numbers of hidden layers, layer widths, and input masking percentages. Each configuration was trained on 40,000 images from the Cifar-10 dataset using stochastic gradient descent (SGD) for network optimisation.

Dataset Range	Accuracy	Top 2	Top 3
0-25	49.05%	70.55%	81.88%
25-50	50.26%	71.13%	82.35%
50-75	50.13%	71.40%	81.91%
75-100	50.04%	70.37%	81.29%

(b) Classification accuracy for the simplified downstream classifier using a masking percentage of 62.5%. Each classifier was trained on 10,000 images from the Cifar-10 dataset (40,000 images split between 4 models) using the ADAM optimiser. The dataset range is the start and end percentage of the dataset (for example 25-50 is the second quarter of the dataset i.e. datapoints 0.25*dataset length through to and including 0.5*dataset length). Top 2 and Top 3 accuracies are the classification accuracy if the correct class is within the top 2 or top 3 probabilities returned by the model respectivly.

Table 5.5: Results of the simplified downstream classifier without federation on the Cifar-10 dataset.

5.3.2 Transfer Learning Experiments

Each masked autoencoder was initially trained on a distinct subset of 60,000 images from the ImageNet dataset [108] for 100 epochs. Following this pre-training, the autoencoders underwent additional fine-tuning on their respective subsets of the Cifar-10 dataset for another 100 epochs. Once the training of the masked autoencoders was complete, the corresponding downstream models were trained on the same Cifar-10 subsets that their associated autoencoders had been fine-tuned on (Table 5.7). This multi-phase training approach aimed to leverage the generalization power of ImageNet pre-training while refining the models for the specifics of the Cifar-10 dataset.

Range	MSE_ℓ	Accuracy	Top 2	Top 3
0-25	1.3641	51.65%	72.42%	83.47%
25-50	1.3518	51.51%	72.95%	83.45%
50-75	1.3617	51.50%	72.13%	82.86%
75-100	1.3545	51.72%	72.27%	82.86%

Table 5.6: Results after 5 federation rounds of 25 epochs each, with 25 pre-federated epochs on Cifar-10. We include a dropout rate of 70% after the single hidden layer

Range	epochs	Transfer Type	Accuracy	Top 2	Top 3
Non-Federated					
0-100	100	None	60.99%	79.69%	88.14%
0-100	100	Encoder Only	61.57%	79.42%	88.07%
0-100	100	Full	61.02%	78.99%	87.56%
Federated, four participants					
0-25	25	Encoder Only	50.72%	72.02%	82.93%
25-50	25	Encoder Only	50.81%	71.28%	82.45%
50-75	25	Encoder Only	49.57%	70.61%	81.51%
75-100	25	Encoder Only	49.80%	71.04%	82.18%
Federated, two participants					
0-50	25	Encoder Only	54.14%	74.55%	84.84%
50-100	25	Encoder Only	54.34%	75.08%	84.37%

Table 5.7: Impact of transfer learning strategies on downstream classifier accuracy for Cifar-10: The masked autoencoder (MAE) was pre-trained on ImageNet, and transfer learning was applied by freezing all weights except for the encoder transformer (Encoder Only) before further training on Cifar-10. This approach outperformed transfer learning without freezing any weights (Full), with both methods yielding superior results compared to training without transfer learning (None).

To enhance the performance of the federated system, several optimizations were implemented. First, transfer learning was applied by freezing all weights of the masked autoencoder except for those in the encoder transformer, allowing participants to benefit from pre-trained knowledge while maintaining adaptability for specific tasks. A single full pre-activation residual bottleneck layer (Figure 5.5) was introduced before the global averaging layer. This layer downsamples the input by a factor of $\frac{1}{8}$ before restoring the original input shape, adding depth and complexity to the network while keeping computational demands reasonable. Notably, the batch normalization layer, already included in the residual bottleneck layer, rendered an additional batch normalization step redundant. Additionally, a dropout rate of 70% was applied after the hidden layer to mitigate overfitting in the federated setting.

Lastly, to further improve results, the encoder transformers of all participants were aggregated

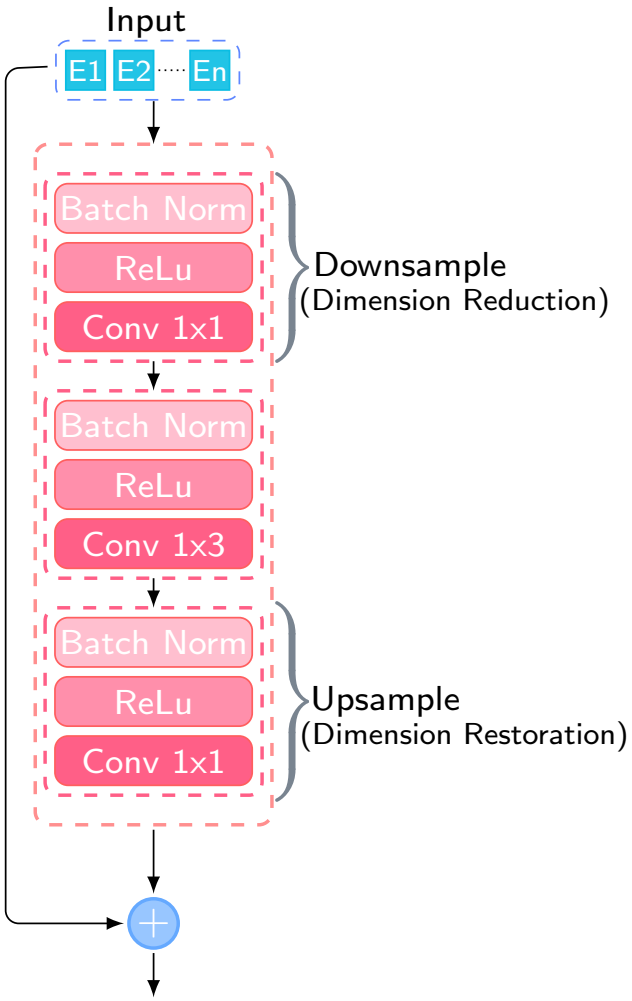


Figure 5.5: The structure of the residual bottleneck layer: The layer consists of three sequential sub-layers. First, a 1D convolution is applied to reduce the input size, with the number of features reduced to $\frac{1}{8}$ of the original. Next, the second sub-layer performs 1D convolution with a kernel size of 3, maintaining the reduced feature dimensions. Finally, the third sub-layer restores the feature dimensions to their original size using a 1D convolution restoring the number of features to its original size. The output of this sequence is then added to the original input, creating a residual connection to enhance feature learning.

prior to the first federated aggregation step. This approach retained the obfuscation capability of the masked autoencoders while significantly improving downstream performance (Table 5.8). The combination of these changes not only enhanced the system's robustness and accuracy but also maintained its privacy-preserving properties, ensuring it remained practical for healthcare IoT applications.

While there is no state-of-the-art to compare these results to, as there has been no research into using masked autoencoders (MAE) in this way, state-of-the-art results for masked autoencoders (MAE) have been achieved using large models, such as ViT-Base, viT-Large and ViT-Huge,

Range	MSE_ℓ	Accuracy	Top 2	Top 3
0-25	1.2852	56.36%	75.73%	85.73%
25-50	1.3150	54.85%	74.93%	84.56%
50-75	1.3349	55.31%	74.82%	84.57%
75-100	1.2862	56.18%	76.03%	85.81%
0-50	1.1971	60.43%	79.15%	87.63%
50-100	1.2444	60.03%	78.76%	87.35%

Table 5.8: Cifar-10 classification accuracy. Each participant's downstream classifier is trained for 25 epochs and aggregated. Thereafter each participant is trained for 5 epochs before aggregation for 25 rounds total.

which obtain classification accuracies between 70.5% and 88.3% on the iNat 2017-2019 datasets, and between 57.0% and 66.8% on the Places205/365 datasets [109]. However, these results are not directly comparable to the research in this chapter for two main reasons:

- Model Size and Deployment Context: The models in these studies are far larger and are designed to run on high-performance hardware. In contrast, the approach in this chapter targets IoT/Edge devices, which have significantly limited computational resources.

- Input Masking Strategy: The state-of-the-art MAE implementations do not apply input masking to the encoder, a central mechanism of this method. This fundamental difference in methodology means that their performance metrics cannot be fairly compared to this research.

In summary, while the reported state-of-the-art results highlight the potential of large transformer models, they do not provide a valid benchmark for this work, which is specifically tailored for resource-constrained IoT/Edge environments and employs a different input processing strategy. However, they are a useful benchmark in predicting what may be possible in the future as IoT/Edge devices become more capable of running large transformers of this type.

Range	Epochs	Rounds	MSE_ℓ	Accuracy	Recall	Precision
Non-Federated						
0-100	100	N/A	1.1354	83.81%	96.67%	81.08%
0-25	100	N/A	0.8677	83.81%	97.44%	80.68%
25-50	100	N/A	1.6054	80.61%	98.97%	76.74%
50-75	100	N/A	1.4703	83.49%	98.97%	79.59%
75-100	100	N/A	0.6741	87.02%	93.33%	86.87%
0-100	25	N/A	0.9001	82.05%	97.69%	78.72%
0-25	25	N/A	0.7228	81.73%	97.95%	78.28%
25-50	25	N/A	0.7187	81.89%	97.18%	78.79%
50-75	25	N/A	0.9720	80.61%	98.72%	76.85%
75-100	25	N/A	0.7121	82.05%	96.67%	79.20%
Federated						
0-25	5	15	0.5253	84.62%	87.95%	87.50%
25-50	5	15	0.8450	81.57%	95.38%	79.32%
50-75	5	15	0.5370	89.58%	94.62%	89.35%
75-100	5	15	0.5533	88.78%	95.13%	87.91%

Table 5.9: Binary classification accuracy of the downstream classifier on the PneumoniaMNIST dataset: Each participant utilized a unique masked autoencoder (MAE) trained independently on distinct subsets of the ChestMNIST dataset. The encoder transformers were not federated, ensuring unique representations for each participant and transfer learning was not required, highlighting the effectiveness of domain-specific training. The effect of dropout on the classifiers is shown in table 5.10.

5.3.3 Chest and Pneumonia Mnist Experiments

While Cifar-10 is a popular benchmark dataset for computer vision tasks, its characteristics differ significantly from those of medical images, both semantically and visually. Unlike conventional digital images, medical scans such as CT (Computed Tomography) and MRI (Magnetic Resonance Imaging) require specialized preprocessing to convert raw data into formats suitable for digital viewing and analysis. For instance, CT scans measure Hounsfield Units (HU), which quantify radiation absorption across a range of -1,024 to 3,071. These values must undergo windowing and levelling processes to map the intensities into a standard 8-bit grayscale range of 0-255, enabling them to be visualized and analyzed effectively.

Single-channel medical images, such as grayscale scans, proved advantageous in this context. Unlike multi-channel images, single-channel formats did not exhibit similar preprocessing challenges and did not necessitate federating the encoder transformer at any stage. This retained the highest level of privacy and enhanced resistance to potential attacks. The system's effective-

ness was demonstrated by training masked autoencoders on distinct subsets of the ChestMNIST dataset [110] without requiring transfer learning, as the visual and semantic similarities between ChestMNIST and the downstream task dataset (PneumoniaMNIST [111]) minimized domain adaptation concerns.

For the downstream binary classification task using the relatively small PneumoniaMNIST dataset (5,000 training images), the system avoided federating masked autoencoders, ensuring that each participant maintained a unique encoder transformer. The federation process employed a low epoch count (5 epochs) between aggregation rounds, with a total of 15 rounds. This strategy resulted in federated performance surpassing the non-federated baseline (Table 5.9). The classifier architecture was largely consistent with those used in the Cifar-10 and ImageNet experiments, incorporating a dropout rate of 15% after the hidden layer to reduce overfitting. Both datasets followed the same training range configurations: training on the entire dataset for ranges of 0-100% and on 25% subsets for restricted ranges. Federated networks were pre-trained for 25 epochs prior to the initial aggregation step. This systematic approach underscores the adaptability and robustness of the framework in handling healthcare-specific IoT applications.

5.4 Conclusion

The transformer architecture has long been a staple in natural language processing (NLP) but has only recently gained prominence in vision tasks, with Vision Transformers (ViTs) demonstrating exceptional performance in image classification [32]. Self-supervised learning has emerged as an invaluable tool for pre-training, particularly in the medical field, where labelling data often requires highly skilled experts or is subject to significant delays, as is the case with conditions like anastomotic leaks. These challenges constrain the rate of labelled data collection. By leveraging the self-supervised capabilities of masked autoencoders (MAEs) and continuing to mask inputs when training downstream classifiers, this system enables the secure sharing of datapoints with enhanced privacy, facilitates federated learning-specific blockchain

validation methods, and mitigates the risks associated with model inversion attacks.

In contrast to the previous chapter where federated networks outperformed the non-federated equivalent, multi-channel datasets pose significant challenges due to their increased data requirements. Fully federating the masked autoencoder could alleviate these data limitations but comes at the cost of reduced privacy and diminished resistance to model inversion attacks. A balanced solution was identified by aggregating the weights of each participant's encoder transformer before initiating federated learning. This compromise maintains a balance between privacy, attack resistance, and performance. However, as single-channel images do not face these issues, further investigation is warranted to explore whether multiple MAEs could handle each channel separately or if preprocessing techniques should be adopted for multi-channel datasets. For instance, when Vision Transformers (ViTs) are applied directly for classification, a learnable classification token is prepended to the set of image patches [32], enabling the model to aggregate global contextual information. While this approach allows the encoder to learn task-specific features, it necessitates the use of a labelled dataset during pre-training, potentially limiting its applicability in scenarios with scarce annotated data.

For greyscale images, MAEs have shown promising results when trained in a self-supervised manner on domain-specific, unlabelled datasets, subsequently serving as feature detectors for downstream classifiers trained on smaller labelled datasets. This approach is particularly suited for scenarios demanding high privacy and fidelity, where labelled data is scarce but unlabelled data is more accessible within the same domain.

MAEs also have potential for image generation tasks [112], enabling dataset augmentation in scenarios where domain-specific data is limited. However, image generation introduces new privacy challenges, as the transparency of model weights and architecture on the blockchain exposes the system to potential attacks. Addressing these challenges will require novel approaches to privacy enhancement.

This chapter has demonstrated a comprehensive integration of federated learning and blockchain decentralization through a custom-built consensus mechanism tailored for neural network training at the edge. Together with the advancements discussed in previous chapters, this research

establishes a novel paradigm for training neural networks. It facilitates inter-hospital collaboration, granting access to otherwise inaccessible data in a fully privacy-preserving manner. Moreover, it harnesses the underutilised IoT and edge devices prevalent in healthcare institutions, enabling the application of artificial intelligence to medical challenges that were previously infeasible due to insufficient or imbalanced datasets.

Range	Epochs	Transfer Type	Dropout	MSE_ℓ	Accuracy	Recall	Precision
Non-Federated							
0-100	25	N/A	N/A	0.8501	82.69	97.18	79.62
0-100	25	N/A	0.1	0.6002	84.13	96.41	81.56
0-100	25	N/A	0.15	0.9001	82.05	97.69	78.72
0-100	25	N/A	0.01	0.5856	83.49	94.36	81.96
0-100	100	N/A	N/A	0.9206	83.81	98.46	80.17
0-100	100	N/A	0.1	0.9853	84.29	98.46	80.67
0-100	100	N/A	0.15	1.1354	83.81	96.67	81.08
0-100	100	N/A	0.01	1.0312	83.65	97.95	80.25
Federated							
0-25	25	N/A	N/A	0.7077	82.05	97.95	78.60
0-25	25	N/A	0.1	0.5826	83.65	96.67	80.90
0-25	25	N/A	0.15	0.7228	81.73	97.95	78.28
0-25	25	N/A	0.01	0.5823	83.81	96.92	80.94
0-25	100	N/A	N/A	0.8106	85.58	94.62	84.25
0-25	100	N/A	0.1	0.7670	85.10	93.59	84.30
0-25	100	N/A	0.15	0.8677	83.81	97.44	80.68
0-25	100	N/A	0.01	0.9761	83.81	98.21	80.29
25-50	25	N/A	N/A	0.7231	81.41	97.18	78.31
25-50	25	N/A	0.1	0.8402	79.81	96.92	76.83
25-50	25	N/A	0.15	0.7187	81.89	97.18	78.79
25-50	25	N/A	0.01	0.8263	79.97	96.92	76.99
25-50	100	N/A	N/A	1.1801	85.26	96.41	82.82
25-50	100	N/A	0.1	1.1594	85.10	96.67	82.49
25-50	100	N/A	0.15	1.6054	80.61	98.97	76.74
25-50	100	N/A	0.01	1.7143	78.85	97.95	75.49
50-75	25	N/A	N/A	0.9830	80.77	98.97	76.89
50-75	25	N/A	0.1	0.7839	84.46	98.46	80.84
50-75	25	N/A	0.15	0.9720	80.61	98.72	76.85
50-75	25	N/A	0.01	0.7943	83.97	98.72	80.21
50-75	100	N/A	N/A	1.1906	83.33	98.72	79.55
50-75	100	N/A	0.1	1.4501	81.73	98.72	77.94
50-75	100	N/A	0.15	1.4703	83.49	98.97	79.59
50-75	100	N/A	0.01	1.4572	84.29	99.23	80.29
75-100	25	N/A	N/A	0.7392	81.73	97.18	78.63
75-100	25	N/A	0.1	0.4633	85.26	94.10	84.17
75-100	25	N/A	0.15	0.7121	82.05	96.67	79.20
75-100	25	N/A	0.01	0.4697	85.26	94.10	84.17
75-100	100	N/A	N/A	0.8867	84.29	95.64	82.16
75-100	100	N/A	0.1	0.8265	86.22	96.41	83.93
75-100	100	N/A	0.15	0.6741	87.02	93.33	86.87
75-100	100	N/A	0.01	0.6322	87.18	93.08	87.26

Table 5.10: The effect of dropout on the Binary classification accuracy of the downstream classifier on the PneumoniaMNIST dataset shown in table 5.9.

Chapter 6

Conclusion

This thesis has explored how artificial neural networks can be effectively trained at the edge by distributing computation across IoT devices for deployment in healthcare applications. The research began by addressing the critical challenge of anastomotic leak detection, demonstrating the necessity of an autonomous detection system while identifying key obstacles: limited, imbalanced data and strict privacy concerns that restrict data sharing between institutions.

To tackle these challenges, a system was developed to generate images of the anastomosis from a 3D model, successfully increasing the amount of training data available for healthcare applications. This approach introduced a novel method for visualising the anastomotic joint, reducing reliance on advanced imaging modalities such as Computed Tomography (CT) or Magnetic Resonance Imaging (MRI). The success of this system underscored the primary hurdle in modern healthcare AI: the scarcity of labelled medical data and the privacy constraints that limit inter-hospital collaboration.

This realisation prompted an investigation into federated learning, a privacy-preserving paradigm enabling collaborative neural network training without the need for sharing sensitive data. To address the lack of trust between participants, blockchain technology was seamlessly integrated, resulting in a decentralised system that required no inherent trust among collaborators. This combination not only ensured robust data security but also demonstrated the ability to outperform traditional network training methods by enabling efficient and secure collaboration across

institutions.

The research culminated in a practical demonstration of federated learning on IoT devices, proving the potential for edge-based neural network training in medical environments. By leveraging IoT devices already present in many hospitals, the feasibility of distributed training at the edge was firmly established. Additionally, the development of a novel blockchain consensus mechanism, tailored specifically for federated learning, fully integrated these technologies and optimised distributed training within existing hospital infrastructure.

By uniting federated learning, IoT edge computing, and blockchain technology, this thesis presented a comprehensive, privacy-preserving framework. This framework addresses the core challenges of data scarcity, privacy, and computational efficiency in modern healthcare, paving the way for broader adoption of AI-driven solutions in clinical practice.

6.1 Summary of Thesis Achievements

This thesis has delivered several significant contributions to the fields of healthcare, machine learning, and IoT, showcasing the potential of decentralised federated learning at the edge. The key achievements are:

- Introduced a novel system capable of generating alternative views of an anastomotic ring from a 3D model, providing a practical substitute for complex imaging modalities such as CT and MRI. This advancement enhances data availability and introduces innovative visualisation methods for clinical applications.
- Produced robust evidence that decentralised federated learning at the edge can outperform traditional, non-federated training approaches.
- Successfully implemented a federated learning system on IoT devices, delivering physical results that demonstrate the feasibility of distributed training in real-world medical environments. This framework is extendable, enabling researchers to replicate and test their own networks using the same infrastructure.

- Designed Proof of Reasoning (PoR), a novel blockchain consensus mechanism specifically tailored for decentralised federated learning. PoR ensures secure and privacy-preserving collaboration between hospitals, addressing the trust deficit in inter-institutional data sharing.

- Enabled advanced federated aggregation by combining MAEs with the PoR consensus mechanism. This approach allows securely encoded data to be shared without imposing constraints on the network architecture, balancing data privacy and utility.

These achievements collectively represent a comprehensive framework for leveraging federated learning, IoT edge computing, and blockchain technologies to address critical challenges in healthcare AI. This work lays the foundation for scalable, privacy-preserving, and collaborative solutions to advance clinical decision-making and medical research.

6.2 Future Work

The research presented in this thesis opens several avenues for further exploration and advancement. Key areas for potential future work include:

- Expanding the Use of Masked Autoencoders (MAEs): Investigating the impact of multi-channel input data on the performance of MAEs for feature extraction could deepen our understanding of their capabilities, particularly in medical imaging.

- Enhancing Blockchain Integration: Incorporating additional blockchain technologies, such as smart contracts, could facilitate automated and trustless operations, improving scalability and efficiency in federated learning frameworks.

- Integrating Generative Capabilities into Classifiers: A natural progression of this work is to integrate generative capabilities into the downstream classifier, discussed in chapter 5. Recent research has shown the potential of combining image generation and classification tasks within a single model, utilising masked autoencoders (MAEs), leading to improved

performance for both tasks [112]. Leveraging this approach could result in more versatile networks that unify generative and discriminative capabilities.

- Autonomous System Design with Bootstrapping: By combining the generative and classification components explored in this thesis, a novel paradigm could be developed where iterative network generations are used to bootstrap subsequent iterations. This would form a self-improving, autonomous learning system capable of advancing without external supervision.

- Federated Generative Adversarial Networks (GANs): GANs present an intriguing opportunity for federated design. Participants could collaboratively train generators against a shared global discriminator, using blockchain-based mechanisms to validate and rank generator performance. Conversely, individual participants could train their discriminators against a federated generator, removing the need to share real data and maintaining privacy.

- Harmonising Generative and Federated Systems: Combining the generative capabilities of GANs with federated learning and blockchain technologies could lead to a more integrated and harmonious framework. This would enable a fully autonomous, privacy-preserving system with both generative and classification capabilities, transforming how Artificial Intelligence (AI) is leveraged in healthcare.

These directions align naturally with the foundational work in this thesis, building upon its contributions to create increasingly sophisticated and impactful systems for healthcare applications. They represent exciting possibilities for advancing federated learning and IoT-enabled Artificial Intelligence (AI) in medical environments.

Bibliography

[1] J. Cafasso, "What is anastomosis?" 18/09/2018 2018. [Online]. Available: https://www.healthline.com/health/anastomosis

[2] A. H. Fang, W. Chao, and M. Ecker, "Review of colonic anastomotic leakage and prevention methods," *Journal of clinical medicine*, vol. 9, no. 12, p. 4061, 2020, 33339209[pmid] PMC7765607[pmcid] jcm9124061[PII]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33339209https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7765607/

[3] TeachMeSurgery, "Anastomotic leak," 17/08/2019 2019. [Online]. Available: https://teachmesurgery.com/perioperative/gastrointestinal/anastomotic-leak/

[4] S. Innovations, "Lumeneye® x1 system," n.d. [Online]. Available: https://surgease.com/our-products/

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, p. 84–90, 2017. [Online]. Available: https://doi.org/10.1145/3065386

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, Conference Proceedings, pp. 770–778.

[7] P. A.-O. Rajpurkar, J. A.-O. Irvin, R. L. Ball, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. A.-O. Langlotz, B. N. Patel, K. A.-O. Yeom, K. A.-O. Shpanskaya, F. G. Blankenberg, J. Seekins, T. A.-O. Amrhein, D. A. Mong, S. A.-O. X. Halabi, E. J.

Zucker, A. Y. Ng, and M. P. Lungren, "Deep learning for chest radiograph diagnosis: A retrospective comparison of the chexnext algorithm to practicing radiologists," *PLOS Medicine*, vol. 15, no. 1549-1676 (Electronic), pp. 1–17, 2018. [Online]. Available: https://doi.org/10.1371/journal.pmed.1002686

[8] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[9] E. Tanghatari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Distributing dnn training over iot edge devices based on transfer learning," *Neurocomputing*, vol. 467, pp. 56–65, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221014235

[10] J. Deng, W. Dong, R. Socher, L. J. Li, L. Kai, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, Conference Proceedings, pp. 248–255.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, Conference Proceedings, pp. 234–241.

[12] J. Konečný, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575*, 2015.

[13] A. Brecko, E. Kajati, J. Koziorek, and I. Zolotova, "Federated learning for edge computing: A survey," *Applied Sciences*, vol. 12, no. 18, p. 9124, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/18/9124

[14] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," 20–22 April 2017.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[16] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, Conference Proceedings, pp. 2107–2116.

[17] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby, "Self-supervised gans via auxiliary rotation loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, Conference Proceedings, pp. 12 154–12 163.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, Conference Proceedings, pp. 1125–1134.

[19] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, Conference Proceedings, pp. 2223–2232.

[20] C.-C. Hsu, C.-W. Lin, W.-T. Su, and G. Cheung, "Sigan: Siamese generative adversarial network for identity-preserving face hallucination," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6225–6236, 2019.

[21] M. Amodio and S. Krishnaswamy, "Travelgan: Image-to-image translation by transformation vector learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, Conference Proceedings, pp. 8983–8992.

[22] M. Sarmad, H. J. Lee, and Y. M. Kim, "Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, Conference Proceedings, pp. 5898–5907.

[23] A. Saeed, F. D. Salim, T. Ozcelebi, and J. Lukkien, "Federated self-supervised learning of multisensor representations for embedded intelligence," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1030–1040, 2021.

[24] X. Chen, C. Lian, L. Wang, H. Deng, S. H. Fung, D. Nie, K.-H. Thung, P.-T. Yap, J. Gateno, J. J. Xia, and D. Shen, "One-shot generative adversarial learning for mri segmentation of craniomaxillofacial bony structures," *IEEE Transactions on Medical Imaging*, vol. 39, no. 3, pp. 787–796, 2020.

[25] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, Conference Proceedings, pp. 7354–7363.

[26] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[27] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," pp. 214–223, 2017. [Online]. Available: https://proceedings.mlr.press/v70/arjovsky17a.html

[28] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are gans created equal? a large-scale study," in *Advances in neural information processing systems*, 2018, Conference Proceedings, pp. 700–709.

[29] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Representation learning and adversarial generation of 3d point clouds," *35th International Conference on Machine Learning (ICML), 2018*, 2017.

[30] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, and S. Gelly, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[33] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, "Unetr: Transformers for 3d medical image segmentation," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2022, Conference Proceedings, pp. 574–584.

[34] Y. Li, S. Rao, J. R. A. Solares, A. Hassaine, R. Ramakrishnan, D. Canoy, Y. Zhu, K. Rahimi, and G. Salimi-Khorshidi, "Behrt: transformer for electronic health records," *Scientific reports*, vol. 10, no. 1, p. 7155, 2020.

[35] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.

[36] A. A. Abdellatif, A. Mohamed, and C. Chiasserini, "Automated class-based compression for real-time epileptic seizure detection," in *2018 Wireless Telecommunications Symposium (WTS)*, 2018, Conference Proceedings, pp. 1–6.

[37] A. Emam, A. A. Abdellatif, A. Mohamed, and K. A. Harras, "Edgehealth: An energy-efficient edge-based remote mhealth monitoring system," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, Conference Proceedings, pp. 1–7.

[38] M. Merenda, M. Astrologo, D. Laurendi, V. Romeo, and F. G. D. Corte, "A novel fitness tracker using edge machine learning," in *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*, 2020, Conference Proceedings, pp. 212–215.

[39] TensorFlow, "Tensorflow for mobile & IoT overview," n.d. [Online]. Available: https://www.tensorflow.org/lite

[40] J. J. López Escobar, R. P. Díaz Redondo, and F. Gil-Castiñeira, "In-depth analysis and open challenges of mist computing," *Journal of Cloud Computing*, vol. 11, no. 1, p. 81, 2022. [Online]. Available: https://doi.org/10.1186/s13677-022-00354-x

[41] E. Chung, J. Fowers, K. Ovtcharov, M. Papamichael, A. Caulfield, T. Massengill, M. Liu, M. Ghandi, D. Lo, S. Reinhardt, S. Alkalay, H. Angepat, D. Chiou,

A. Forin, D. Burger, L. Woods, G. Weisz, M. Haselman, and D. Zhang, "Serving dnns in real time at datacenter scale with project brainwave," *IEEE Micro*, vol. 38 ,, pp. 8–20, 2018. [Online]. Available: https://www.microsoft.com/en-us/research/publication/serving-dnns-real-time-datacenter-scale-project-brainwave/

[42] J. Fowers, K. Ovtcharov, M. Papamichael, T. Massengill, M. Liu, D. Lo, S. Alkalay, M. Haselman, L. Adams, M. Ghandi, S. Heil, P. Patel, A. Sapek, G. Weisz, L. Woods, S. Lanka, S. Reinhardt, A. Caulfield, E. Chung, and D. Burger, "A configurable cloud-scale dnn processor for real-time ai," 1–6 June 2018 2018. [Online]. Available: https://www.microsoft.com/en-us/research/publication/a-configurable-cloud-scale-dnn-processor-for-real-time-ai/

[43] A. Afroj, Q. Sahar, I. Naiyar, and R. Khalid, *Fog, Edge and Pervasive Computing in Intelligent Internet of Things Driven Applications in Healthcare: Challenges, Limitations and Future Use*. Piscataway, NY, USA: IEEE, 2021, pp. 1–26. [Online]. Available: http://ieeexplore.ieee.org/document/9292565

[44] N. K. Giang, R. Lea, M. Blackstock, and V. C. M. Leung, "Fog at the edge: Experiences building an edge computing platform," in *2018 IEEE International Conference on Edge Computing (EDGE)*, 2018, Conference Proceedings, pp. 9–16.

[45] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: Progress and challenges," in *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, 2016, Conference Proceedings, pp. 83–84.

[46] G. Labrèche, D. Evans, D. Marszk, T. Mladenov, V. Shiradhonkar, T. Soto, and V. Zelenevskiy, "Ops-sat spacecraft autonomy with tensorflow lite, unsupervised learning, and online machine learning," in *2022 IEEE Aerospace Conference (AERO)*, 2022, Conference Proceedings, pp. 1–17.

[47] P. Warden; and D. Situnayake, *TinyML*. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2019.

[48] A. Rodriguez;, W. Li;, J. Dai;, F. Zhang;, J. Gong;, and C. Yu, "Intel processors for deep learning training," 22/03/2018 2017. [Online]. Available: https://software.intel.com/content/www/us/en/develop/articles/intel-processors-for-deep-learning-training.html

[49] W. Hong, J. Meng, and J. Yuan, "Distributed composite quantization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11294

[50] S. Latif, R. Rana, J. Qadir, A. Ali, M. A. Imran, and M. S. Younis, "Mobile health in the developing world: Review of literature and lessons from a case study," *IEEE Access*, vol. 5, pp. 11 540–11 556, 2017.

[51] J. Minoi, M. R. Suhaili, and A. W. Yeo, "A holistic ecosystem for rural mhealth applications and lesson learnt," in *2014 IEEE Conference on Biomedical Engineering and Sciences (IECBES)*, 2014, Conference Proceedings, pp. 1003–1008.

[52] H. K. Lim, J. B. Kim, S. Y. Kim, and Y. H. Han, "Federated reinforcement learning for automatic control in sdn-based iot environments," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, Conference Proceedings, pp. 1868–1873.

[53] S. Oueida, Y. Kotb, M. Aloqaily, Y. Jararweh, and T. Baker, "An edge computing based smart healthcare framework for resource management," *Sensors*, vol. 18, p. 4307, 2018.

[54] D. Lin and Y. Tang, "Edge computing-based mobile health system: Network architecture and resource allocation," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1716–1727, 2020.

[55] W. Bao, C. Wu, S. Guleng, J. Zhang, K. L. A. Yau, and Y. Ji, "Edge computing-based joint client selection and networking scheme for federated learning in vehicular iot," *China Communications*, vol. 18, no. 6, pp. 39–52, 2021.

[56] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

[57] N. K. Giang, V. C. Leung, and R. Lea, "On developing smart transportation applications in fog computing paradigm," p. 91–98, 2016. [Online]. Available: https://doi.org/10.1145/2989275.2989286

[58] S. S. Hajam and S. A. Sofi, "Iot-fog architectures in smart city applications: A survey," *China Communications*, vol. 18, no. 11, pp. 117–140, 2021.

[59] N. Subramanian, S. M. G.B, J. P. Martin, and K. Chandrasekaran, "Htmrpl++ : A trust-aware rpl routing protocol for fog enabled internet of things," in *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2020, Conference Proceedings, pp. 1–5.

[60] STMicroelectronics, "Stm32 solutions for artificial neural networks," n.d. [Online]. Available: https://www.st.com/content/st_com/en/stm32-ann.html

[61] S. Augenstein, H. B. McMahan, D. Ramage, S. Ramaswamy, P. Kairouz, M. Chen, and R. Mathews, "Generative models for effective ml on private, decentralized datasets," *arXiv preprint arXiv:1911.06679*, 2019.

[62] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *arXiv preprint arXiv:1911.07963*, 2019.

[63] Z. Yang, Y. Shi, Y. Zhou, Z. Wang, and K. Yang, "Trustworthy federated learning via blockchain," *IEEE Internet of Things Journal*, vol. 10, no. 1, pp. 92–109, 2022.

[64] A. Islam, A. A. Amin, and S. Y. Shin, "Fbi: A federated learning-based blockchain-embedded data accumulation scheme using drones for internet of things," *IEEE Wireless Communications Letters*, vol. 11, no. 5, pp. 972–976, 2022.

[65] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC 2006*, ser. Theory of Cryptography. Springer Berlin Heidelberg, 2006, Conference Proceedings, pp. 265–284.

[66] C. Dwork, "Differential privacy," in *ICALP 2006*, ser. Automata, Languages and Programming. Springer Berlin Heidelberg, 2006, Conference Proceedings, pp. 1–12.

[67] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, Conference Proceedings, pp. 308–318.

[68] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *ASIACRYPT 2017*, ser. Advances in Cryptology – ASIACRYPT 2017. Springer International Publishing, 2017, Conference Proceedings, pp. 409–437.

[69] P. Rizomiliotis and A. Triakosia, "On matrix multiplication with homomorphic encryption," p. 53–61, 2022. [Online]. Available: https://doi.org/10.1145/3560810.3564267

[70] J. Xiong, J. Chen, J. Lin, D. Jiao, and H. Liu, "Enhancing privacy-preserving machine learning with self-learnable activation functions in fully homomorphic encryption," *Journal of Information Security and Applications*, vol. 86, p. 103887, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212624001893

[71] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *TCC 2011*, ser. Theory of Cryptography. Springer Berlin Heidelberg, 2011, Conference Proceedings, pp. 253–273.

[72] M. S. Thomas and D. A. Margolin, "Management of colorectal anastomotic leak," *Clinics in colon and rectal surgery*, vol. 29, no. 2, pp. 138–144, 2016, 27247539[pmid] PMC4882170[pmcid] 00728[PII]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/27247539https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4882170/

[73] U. A. Dietz and E.-S. Debus, "Intestinal anastomoses prior to 1882; a legacy of ingenuity, persistence, and research form a foundation for modern gastrointestinal surgery," *World Journal of Surgery*, vol. 29, no. 3, pp. 396–401, 2005. [Online]. Available: https://doi.org/10.1007/s00268-004-7720-x

[74] A. L. Peel and E. W. Taylor, "Proposed definitions for the audit of postoperative infection: a discussion paper. surgical infection study group," *Annals of the Royal College of Surgeons of England*, vol. 73, no. 6, pp. 385–388, 1991, 1759770[pmid] PMC2499458[pmcid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/1759770https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2499458/

[75] R. Kennedy, I. Jenkins, and P. J. Finan, "Controversial topics in surgery: Splenic flexure mobilisation for anterior resection performed for sigmoid and rectal cancer," *Annals of the Royal College of Surgeons of England*, vol. 90, no. 8, pp. 638–642, 2008, 18990277[pmid] PMC2727804[pmcid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/18990277https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2727804/

[76] N. Hyman, T. L. Manchester, T. Osler, B. Burns, and P. A. Cataldo, "Anastomotic leaks after intestinal anastomosis: it's later than you think," *Annals of surgery*, vol. 245, no. 2, pp. 254–258, 2007, 17245179[pmid] PMC1876987[pmcid] 00000658-200702000-00014[PII]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/17245179https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1876987/

[77] J. G. Docherty, J. R. McGregor, A. M. Akyol, G. D. Murray, and D. J. Galloway, "Comparison of manually constructed and stapled anastomoses in colorectal surgery. west of scotland and highland anastomosis study group," *Annals of surgery*, vol. 221, no. 2, pp. 176–184, 1995, 7857145[pmid] PMC1234951[pmcid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/7857145https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1234951/

[78] A. Fingerhut, A. Elhadad, J. M. Hay, F. Lacaine, and Y. Flamant, "Infraperitoneal colorectal anastomosis: hand-sewn versus circular staples. a controlled clinical trial. french associations for surgical research," *Surgery*, vol. 116, no. 3, pp. 484–490, 1994. [Online]. Available: http://europepmc.org/abstract/MED/8079178

[79] A. Fingerhut, J. M. Hay, A. Elhadad, F. Lacaine, and Y. Flamant, "Supraperitoneal colorectal anastomosis: hand-sewn versus circular staples–a controlled clinical trial.

french associations for surgical research," *Surgery*, vol. 118, no. 3, pp. 479–485, 1995. [Online]. Available: http://europepmc.org/abstract/MED/7652682https://doi.org/10.1016/s0039-6060(05)80362-9

[80] E. L. Bokey, P. H. Chapuis, C. Fung, W. J. Hughes, S. G. Koorey, D. Brewer, R. C. Newland, and Y. S. Y. Chiu, "Postoperative morbidity and mortality following resection of the colon and rectum for cancer," *Diseases of the Colon & Rectum*, vol. 38, no. 5, pp. 480–487, 1995. [Online]. Available: https://doi.org/10.1007/BF02148847

[81] F. Goulder, "Bowel anastomoses: The theory, the practice and the evidence base," *World journal of gastrointestinal surgery*, vol. 4, no. 9, pp. 208–213, 2012, 23293735[pmid] PMC3536859[pmcid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/23293735https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3536859/

[82] L. Fielding, S. Stewart-Brown, L. Blesovsky, and G. Kearney, "Anastomotic integrity after operations for large-bowel cancer: a multicentre study," *Br Med J*, vol. 281, no. 6237, pp. 411–414, 1980.

[83] K. G. Walker, S. W. Bell, M. J. F. X. Rickard, D. Mehanna, O. F. Dent, P. H. Chapuis, and E. L. Bokey, "Anastomotic leakage is predictive of diminished survival after potentially curative resection for colorectal cancer," *Annals of surgery*, vol. 240, no. 2, pp. 255–259, 2004, 15273549[pmid] PMC1356401[pmcid] 00000658-200408000-00010[PII]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/15273549https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1356401/

[84] C. S. McArdle, D. C. McMillan, and D. J. Hole, "Impact of anastomotic leakage on long-term survival of patients undergoing curative resection for colorectal cancer," *British Journal of Surgery*, vol. 92, no. 9, pp. 1150–1154, 2005. [Online]. Available: https://doi.org/10.1002/bjs.5054

[85] J. Pickleman, W. Watson, J. Cunningham, S. G. Fisher, and R. Gamelli, "The failed gastrointestinal anastomosis: an inevitable catastrophe?" *Journal of the*

American College of Surgeons, vol. 188, no. 5, pp. 473–482, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1072751599000289

[86] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision – ECCV 2016*, ser. Computer Vision – ECCV 2016. Springer International Publishing, 2016, Conference Proceedings, pp. 630–645.

[87] A. Dhere and J. Sivaswamy, "Self-supervised learning for segmentation," *arXiv preprint arXiv:2101.05456*, 2021.

[88] K. Foundation, "Krita," 2022. [Online]. Available: https://krita.org/en

[89] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," pp. 580–587, 2014.

[90] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," p. 1097–1105, 2012.

[91] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[92] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," June 2015.

[93] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[94] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010, Conference Proceedings.

[95] Y. Tudela, M. Majó, N. de la Fuente, A. Galdran, A. Krenzer, F. Puppe, A. Yamlahi, T. N. Tran, B. J. Matuszewski, K. Fitzgerald, C. Bian, J. Pan, S. Liu, G. Fernández-Esparrach, A. Histace, and J. Bernal, "A complete benchmark for polyp detection, segmentation and classification in colonoscopy images," *Front Oncol*, vol. 14, p. 1417862, 2024, 2234-943x

Tudela, Yael Majó, Mireia de la Fuente, Neil Galdran, Adrian Krenzer, Adrian Puppe, Frank Yamlahi, Amine Tran, Thuy Nuong Matuszewski, Bogdan J Fitzgerald, Kerr Bian, Cheng Pan, Junwen Liu, Shijle Fernández-Esparrach, Gloria Histace, Aymeric Bernal, Jorge Journal Article Switzerland 2024/10/09 Front Oncol. 2024 Sep 24;14:1417862. doi: 10.3389/fonc.2024.1417862. eCollection 2024.

[96] B. Foundation, "Blender," 2022. [Online]. Available: https://www.blender.org/

[97] G. F.-E. J. Bernal, F. J. Sánchez and C. Rodríguez, "Cvc-clinicdb," 2015.

[98] M. Ishaq, M. H. Afzal, S. Tahir, and K. Ullah, "A compact study of recent trends of challenges and opportunities in integrating internet of things (iot) and cloud computing," in *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, 2021, Conference Proceedings, pp. 1–4.

[99] S. Vishnu, S. R. J. Ramson, and R. Jegan, "Internet of medical things (iomt) - an overview," in *2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*, 2020, Conference Proceedings, pp. 101–104.

[100] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent advances in the internet-of-medical-things (iomt) systems security," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8707–8718, 2021.

[101] D. R. Brendan McMahan, "Federated learning: Collaborative machine learning without centralized training data," 2017. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[102] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, Conference Proceedings, pp. 1175–1191.

[103] K. Alex and H. Geoffrey, "Cifar-10 (canadian institute for advanced research)," Toronto University, Report, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[104] P. R. Nair and D. R. Dorai, "Evaluation of performance and security of proof of work and proof of stake using blockchain," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, Conference Proceedings, pp. 279–283.

[105] S. A. Y. Chicaiza, C. N. S. Chafla, L. F. E. Álvarez, P. F. I. Matute, and R. D. Rodriguez, "Analysis of information security in the pow (proof of work) and pos (proof of stake)blockchain protocols as an alternative for handling confidential information in the public finance ecuadorian sector," in *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*, 2021, Conference Proceedings, pp. 1–5.

[106] S. Masseport, B. Darties, R. Giroudeau, and J. Lartigau, "Proof of experience: empowering proof of work protocol with miner previous work," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, 2020, Conference Proceedings, pp. 57–58.

[107] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, Conference Proceedings, pp. 16 000–16 009.

[108] P. Chrabaszcz, I. Loshchilov, and F. Hutter, "A downsampled variant of imagenet as an alternative to the cifar datasets," *arXiv preprint arXiv:1707.08819*, 2017.

[109] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, Conference Proceedings, pp. 16 000–16 009.

[110] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, Conference Proceedings, pp. 2097–2106.

[111] D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. L.

Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. N. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131.e9, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0092867418301545

[112] T. Li, H. Chang, S. Mishra, H. Zhang, D. Katabi, and D. Krishnan, "Mage: Masked generative encoder to unify representation learning and image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, Conference Proceedings, pp. 2142–2152.