University of Zurich[UZH]

*Burkhard Stiller, Alberto Huertas, Chao Feng, Daria Schumm,
Jan von der Assen, Katharina O. E. Müller, Nazim Nezhadsistani,
Thomas Grübl, Weijie Niu (Edts.)*

Communication Systems XVII

TECHNICAL REPORT – No. IFI-2024.01

June 2024

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Introduction

The Department of Informatics (IFI) of the University of Zurich, Switzerland works on research and teaching in the area of computer networks and communication systems. Communication systems include a wide range of topics and drive many research and development activities. Therefore, during the spring term FS 2024 a new instance of the Communication Systems seminar has been prepared and students as well as supervisors worked on this topic.

The areas of communication systems include among others wired and wireless network technologies, various network protocols, network management, Quality-of-Service (QoS) provisioning, mobility, security aspects, peer-to-peer systems, multimedia communication, and manifold applications, determining important parts of future networks. Therefore, this year's seminar addressed such areas in more depth. The understanding and clear identification of problems in technical and organizational terms have been prepared and challenges as well as weaknesses of existing approaches have been addressed. All talks in this seminar provide a systematic approach to judge dedicated pieces of systems or proposals and their suitability.

Content

This new edition of the seminar entitled "Communication Systems XVII" discusses a number of selected topics in the area of computer networks and communication systems. Talk 1 begins with an overview of Single Instruction, Multiple Data (SIMD) techniques applied to networking applications, highlighting how SIMD-driven parallelism enhances performance in networking tasks through specific algorithms and implementations. Talk 2 delves into privacy-preserving synthetic data generation, offering a taxonomy of existing techniques and evaluating their effectiveness and challenges in maintaining data privacy. Talk 3 reviews federated learning for large language models (LLMs), discussing the benefits and challenges of this approach, focusing on data privacy, communication efficiency, and model performance. Talk 4 explores inference attacks on machine learning, covering types such as membership inference and model inversion attacks, their implications for data privacy and security, and mitigation strategies. Talk 5 examines the security and privacy issues associated with the Domain Name System (DNS), analyzing common attacks like cache poisoning and DNS spoofing, along with current mitigation strategies. Talk 6 investigates the impact of heterogeneous IoT networks on privacy in smart home environments, discussing how diverse IoT device integration affects data privacy and security, supported by case studies. Finally, Talk 7 explores the potential of blockchain technology to transform financial services, addressing applications such as decentralized finance (DeFi), smart contracts, and secure transactions, along with the benefits and challenges of implementing blockchain solutions in the financial sector.

Seminar Operation

Based on well-developed experiences of former seminars, held in different academic environments, all interested students worked on an initially offered set of papers and book chapters. Those relate to the topic titles as presented in the Table of Content below. They prepared a written essay as a clearly focused presentation, an evaluation, and a summary of those topics. Each of these essays is included in this technical report as a separate section and allows for an overview on important areas of concern, technology architectures and functionality, sometimes business models in operation, and problems encountered.

In addition, every group of students prepared a slide presentation of approximately 45 minutes to present its findings and summaries to the audience of students attending the seminar and other interested students, research assistants, and professors. Following a general question and answer phase, a student-lead discussion debated open issues and critical statements with the audience.

Local IFI support for preparing talks, reports, and their preparation by students had been granted by Christos Tsiaras, Andri Lareida, Lisa Kristiana, Radhika Garg, Daniel Dönni, Corinna Schmitt, and Burkhard Stiller. In particular, many thanks are addressed to Corinna Schmitt for her strong commitment on getting this technical report ready and quickly published. A larger number of pre-presentation discussions have provided valuable insights in the emerging and moving field of communication systems, both for all groups of students and supervisors. Many thanks to all people contributing to the success of this event, which has happened in a lively group of highly motivated and technically qualified students and people.

Zürich, June 2024

Contents

Chapter 1

An Overview and Analysis of SIMD-driven Parallelism in Networking Applications

Maximilian Huwyler, Tobias Frauenfelder

This paper presents an in-depth analysis of SIMD applications in the networking sector with a focus on performance, energy use, and computational complexity. SIMD stands for Single Instruction Multiple Data and is a parallel computing technique that allows the processing of multiple data concurrently, which is often used to speed up repetitive tasks. These are all important factors when it comes to networking applications, which gained more relevance due to the rise of cloud computing. After giving an overview over the basics, this paper covers the network-related topics of bloom filters, longest prefix matching, deep packet inspection, random network coding as well as IoT authentication and security. The literature review hints that SIMD can have a significant impact on the performance of networking applications with some trade-offs with regard to flexibility, due to often required sophisticated data layouts. This raises doubts about the extent of benefits in real-world, practical applications.

Contents

1.1 Introduction and Problem Statement

Parallel computing techniques play an important role in efficiently executing programs and other computations for multiple applications. Next to well-known techniques of multithreading, which makes use of multiple processing cores on a computer chip, there exist also other ways of executing programs in parallel. One of them is making use of Single Instruction Multiple Data (SIMD) instructions, which allow programmers or compilers to execute one instruction on multiple data instances concurrently [19]. This can be useful if there is a necessity to perform the same operation on lots of different elements. Let us for example take a look at two vectors $\mathbf{v}_1 = (1, 2, 3)$ and $\mathbf{v}_2 = (4, 5, 6)$. If we want to add those vectors with scalar operations, we need three instructions, while with SIMD only one is necessary. SIMD can be used in various fields, one of it is in networking applications.

Since cloud computing has become as popular as today, users all over the world rely on various networking applications [6]. It is important to ensure that these run as efficiently as possible to provide fast and reliable service to users. This can be achieved through the use of parallel computing. SIMD helps to reduce the complexity of operations in high-throughput networking environments. Additionally, it has the possibility to increase energy efficiency through using a lower amount of energy. Network applications that can become more efficient through SIMD computations include packet processing, data filtering, and encryption, while this is not an exhaustive list of applications.

While SIMD is already widely used in multimedia applications, research in computer networks is trying to discover how SIMD can be used for benefitting advantages in their field to enhance performance, improve energy efficiency, and decrease the complexity of programs. The aim of this paper is to provide an overview of different research projects in the networking sector and give an overview of where SIMD parallelization can be used. Due to the complexity of this field, we first make sure to cover the background of SIMD to later dive into concrete research topics

This seminar paper will be structured as follows: First, we will take a look at the Background of SIMD applications in Section 5.2. We will look at Flynn's taxonomy and the difference between SISD and SIMD computer architectures. Following, this paper provides a common multimedia application of SIMD called chroma keying. After covering the basics, we will dive into the field of networking applications and take a closer look at selected research papers in Section 1.3. We will cover how SIMD benefits bloom filters (BF), longest prefix matching (LPM), deep packet inspection (DPI), random linear network coding (RLNC), IoT Authentication, and IoT Security. After covering those fields, we will provide a summary and conclusion of this paper in Section 5.6.

1.2 Background

1.2.1 Flynn's Taxonomy

In 1966, Flynn [5] created a taxonomy to classify very high-speed computers into different classes. The taxonomy distinguishes computers according to their capability to processes single or multiple data at the same time. Additionally, it is distinguished if multiple instructions are applied to the piece of data or not. The taxonomy results in four different computer architectures namely, Single Instruction Single Data (SISD), Single Instruction Multiple Data (SIMD), Multiple Instruction Single Data (MISD), and Multiple Instruction Multiple Data (MIMD).

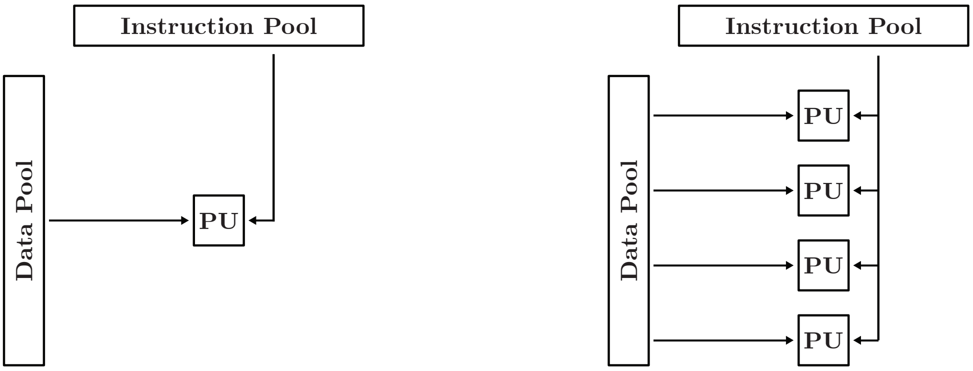
	Single Instruction	**Multiple Instruction**
Single Data	SISD	MISD
Multiple Data	SIMD	MIMD

Table 1.1: Flynn's Taxonomy

To enhance comprehension, it is useful to examine a specific example for every classification. A computer with a SISD architecture is a state-of-the-art processor with a von Neumann architecture. A concrete example of a processor like this is the *Intel 80486* released in 1989. A Computer with a MISD architecture is a computer that is used for safety-critical computation. An explicit instance of a processor like this would be the calculations in a space shuttle. Flight critical programs are executed there by four times simultaneously to ensure redundancy [21]. A computer architecture that makes use of SIMD instructions is useful for computers that have to do the same computation over and over again, like computations related to multimedia. One of the first processors with SIMD capability was the *Intel Pentium MMX* which was released in 1996 due to the higher demand of multimedia computations [19]. The MIMD processor category includes the processors that are installed in most laptops and cell phones nowadays, which are multicore processors. A particular exemplar of a MIMD processor is the *Intel Core i9-14900K* released in 2024.

1.2.2 SISD vs SIMD

To grasp the distinction between SISD (see Fig. 1.1a) and SIMD (see Fig. 1.1b) computer architectures, it is helpful to examine a specific example. Imagine that we have a simple assembly-like program that operates on an array of integers. The program adds one to every element and then multiplies it by two. A program like this is depicted in Figure 1.2.



(a) SISD Architecture

(b) SIMD Architecture

Figure 1.1: Comparison of SISD and SIMD architectures

LD $R \leftarrow A[i]$
ADD $R \leftarrow R + 1$
MUL $R \leftarrow R \times 2$
ST $A[i] \leftarrow R$

Figure 1.2: Sequential operations in assembly language

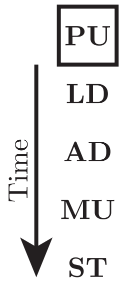


Figure 1.3: Flowchart illustrating the SISD program execution

The program first loads one element of the array into a register, adds one to the value in the register, and multiplies it by two. Once all these steps are executed after each other, the value from the register is stored back into the array. Since all of those operations are independent of each other and therefore can be executed simultaneously, the program is suited for SIMD-capable processors like an array processor depicted in Figure 1.1b.

So that the program can be executed on a processor with SIMD architecture, the Assembly instructions have to be adjusted. As it can be seen in Figure 1.4, the program loads four elements concurrently into a vector register and performs all operations on them at the same time, instead of a single element at the time. This is possible due to the SIMD architecture (depicted in Figure 1.5) which has multiple processing units. SIMD processors can be classified into array and vector processors. They differ from each other in how they operate. Array processors are composed of identical units, whereas vector processors contain a variety of units, each specialized for specific operations. In array processors, operations are performed simultaneously but in different locations, whereas, in vector processors, operations occur sequentially in the same location [15].

LD $VR \leftarrow A[3:0]$
ADD $VR \leftarrow VR + 1$
MUL $VR \leftarrow VR \times 2$
ST $A[3:0] \leftarrow VR$

Figure 1.4: Sequential operations in assembly language

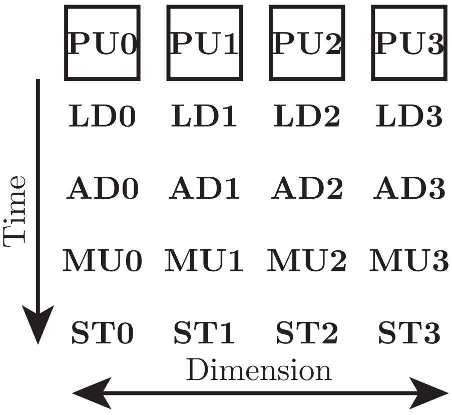


Figure 1.5: Flowchart illustrating the SIMD program execution [15]

1.2.3 Advantages and Disadvantages of SIMD

As shown in Subsection 1.2.2, computers that have SIMD capability can speed up the execution of a program when there are lots of independent operations that can be executed concurrently. This results in a faster program execution compared to one on a processor with a SISD architecture. SIMD processors are nowadays available on most processors today. Smart compilers are also capable to make use of SIMD extensions, although the programmer wrote their program sequentially. However, this is not always possible, as it will be shown in this subsection.

Handling of Conditionals: If we for example consider a program that subtracts element-wise two arrays from each other if the element of the first array is greater than 5, we can see that this program can not be parallelized as easily as the first example presented in this paper, since SIMD does not have a concept of branching. The simple loop described has to be translated into another one that first calculates all subtractions from the array, then calculates all the conditionals, and in the end filters out all the values for which the condition is true. This creates lots of overhead for smaller calculations [26].

Loop Dependencies: If we consider a program that takes an array of integers and adds the previous element to the current one, we can see that this program can not simply be translated into a SIMD-suitable program. Every iteration of this program is dependent on the previous operation. Therefore, those iterations cannot be executed concurrently [11].

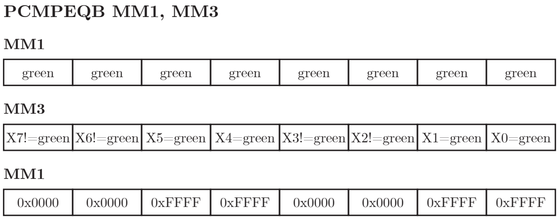
Complex Types: SIMD is capable of processing types like integers, shorts, and floats and in some cases small classes with a few members. However, if a program makes use of more complex types of data structures like binary trees or hash maps, SIMD extensions are not suited [8].

1.2.4 General Field of Application

Although SIMD is nowadays used in Networking Applications, it was first widely used for multimedia use like the processing of images, films, or audio as well as 3D graphics. One of the first widely used processors with SIMD extensions was the *Intel Pentium with MMX Technology*, introduced in 1996. The SIMD instructions on the MMX processor work with floating-point registers. MMX defines 8 floating point registers named from MM0 to MM7. Every register is 64-bit wide, so it can work on 8 bits simultaneously[19].

One example the MMX can be used for is introduced by Peleg and Wiser from Intel is chroma keying [16]. Chroma keying is used to remove the background of an image in the color blue or green and replace it with some other image. This application is well suited for the MMX processor since it needs to do the same task all over again.

The program starts first with creating a bitmask, masking the part of the foreground image that should be kept or removed. To achieve this, the reference color of the background that has to be removed is loaded into the register, as well as eight pixels of the image with the background to remove. The *PCMPEQB* command compares bytes in two MMX registers for equality and creates a bit mask of the image of the men in front of the green screen, which we want to keep.



PCMPEQB MM1, MM3

MM1

green	green	green	green	green	green	green	green

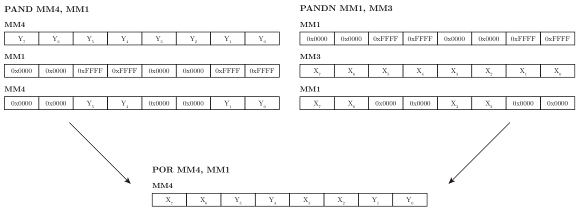
MM3

X7!=green	X6!=green	X5=green	X4=green	X3!=green	X2!=green	X1=green	X0=green

MM1

0x0000	0x0000	0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF

(a) Creation of the Bitmask



PAND MM4, MM1

MM4

Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0

MM1

0x0000	0x0000	0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF

MM4

0x0000	0x0000	Y_5	Y_4	0x0000	0x0000	Y_1	Y_0

PANDN MM1, MM3

MM1

0x0000	0x0000	0xFFFF	0xFFFF	0x0000	0x0000	0xFFFF	0xFFFF

MM3

X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0

MM1

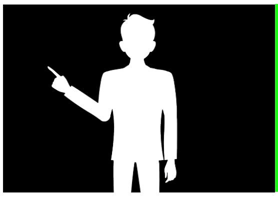
X_7	X_6	0x0000	0x0000	X_3	X_2	0x0000	0x0000

POR MM4, MM1

MM4

X_7	X_6	Y_5	Y_4	X_3	X_2	Y_1	Y_0

(b) Merging Image X and Y to the new image



(c) Bitmask

(d) Image X

(e) Image Y

(f) New Image

Upon creation of the bitmask, the commands *PAND* and *PANDN*, which perform bitwise logical AND/AND NOT operations, select the parts of the images, that we want to keep. In a later step with Packet wise *OR* operation *POR* merges the background and the foreground image to a new image.

1.3 Usage of SIMD in Networking Applications

To conduct an overview of different topics of SIMD use in networking applications, the IEEE Xplore, as well as Google Scholar, were employed. Besides one paper, recent literature that was written in the last six years was searched for to give an overview of the recent development of SIMD processing in computer networking. To paint a picture of how SIMD processing is used, we discuss three applications in great detail: Bloom filters [14] [13], longest prefix matching [22] [23], and deep packet inspection (DPI) [25]. Afterward, SIMD use in random linear network coding (RLNC) [18] [20] is presented. In the end, we show two SIMD applications in the field of IoT authentication and security [3] [12].

1.3.1 Bloom Filters

A bloom filter is a data structure that supports membership checks. Bloom filters are probabilistic, since false positives are possible. An element could not be part of a set, but the bloom filter could identify it as a member. On the other hand, false negatives are not possible. If an element is not part of a set then the bloom filter would never identify it as such. This trade-off enables the bloom filter to be fast and space-efficient. A simple bloom filter implementation consists of an array of zeros and a hash function and represents a set. The two operations that are supported are inserting a new member and checking the membership of an element. For each element that is inserted into the set, the element is hashed onto the array, and corresponding bits are set to one. During a membership check, an element is hashed into the array, and it is checked whether all the bits are already set (see Figure 1.7) [2].

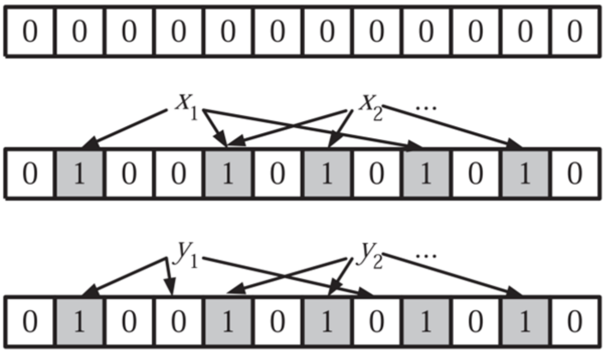


Figure 1.7: Bloom Filter [2]

Bloom filters have a wide variety of applications in high-speed computer networking. The speed of the implementation of bloom filters can be a bottleneck in those situations, and therefore it is beneficial to make use of the available hardware and leverage SIMD architecture to accelerate the membership tests [14] [13].

Lu et al. [14] propose a new Bloom Filter variant called Ultra-Fast Bloom Filters (UFBF). They introduce a basic data structure that consists of words with w bits. k of those words make up a block and the final bit array is made up of r blocks. For now, we assume that

k is the number of membership bits set for each element and w is the length of a general modern CPU register (32- or 64-bit) (see Figure 1.8).

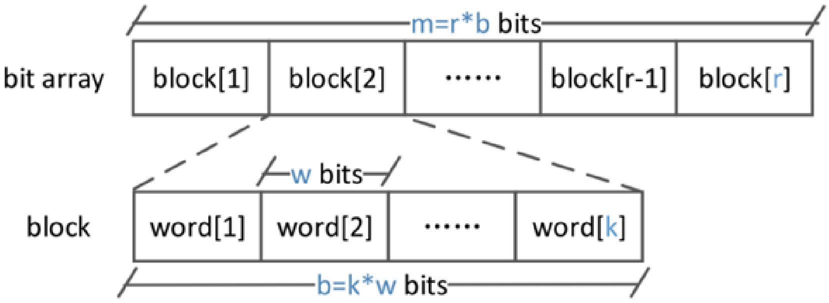


Figure 1.8: Basic Data Structure [14]

To insert an element into the UFBF a hash function is used to map the element onto a single block. For each word in the block, another hash function is used to map the element onto a bit and set this bit (see Figure 1.9). The checking process is analog. The difference is that instead of setting the bits, we see if the bits are checked [14].

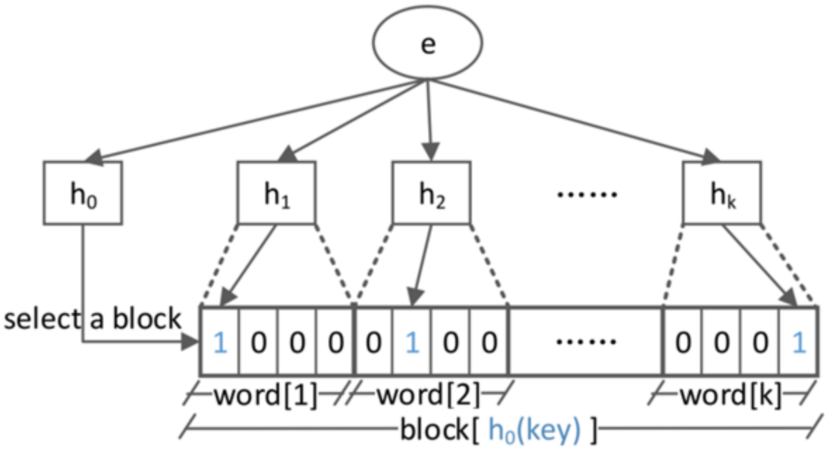


Figure 1.9: Insertion Process [14]

The core idea of UFBF is to parallelize the hash function computation and bit-test process by replacing the ordinary operations with their SIMD equivalent (see Figure 1.20). Assuming our SIMD processor can compute the result of p arithmetic operations in parallel, then p hash values are computed in parallel using the same hash map but different seed values. The membership check parallelism stems from the structure of the data. As mentioned above, we map an element into one block with k consecutive words. These words can be loaded at the same time such that the bit test can be performed in parallel. The mapping onto a single block also improves the cache efficiency, since one block can be chosen to fit into a single cache line [14].

Compared to the standard bloom filter (SBF) the UFBF has a higher false positive rate but less overhead during membership checking. Additionally, this the UFBF suffers applicability because p is fixed for a given SIMD processor and we assume that $k \leq p$. Depending on the application of the bloom filter, we would like to choose k appropriately large. The authors propose a generalization called c-UFBF which works with a higher word number k but is a trade-off between the SBF and the UFBF. During an evaluation, the authors show the trade-off between higher false probability and better membership cache efficiency in a practical setting. They also show that the performance of the c-UFBF lies between the SBF and UFBF. Ultimately, they state that the tradeoff of UBFB is worthwhile in their opinion [14].

Li et al. [13] propose an alternative use of SIMD to accelerate bloom filters. They use a vectorization algorithm where sixteen elements are loaded into a 512-bit vector. A corresponding seed index vector provides the seed that is used to compute the hash of the element. The hashes of the elements can then be tested against the set bits in the bloom filter. This is done several times for each element with different hash values. If an element fails the bit test earlier than the other, a new element is loaded in the key vector, and the seed index vector is adjusted for the newly loaded elements (see Figure 1.10).

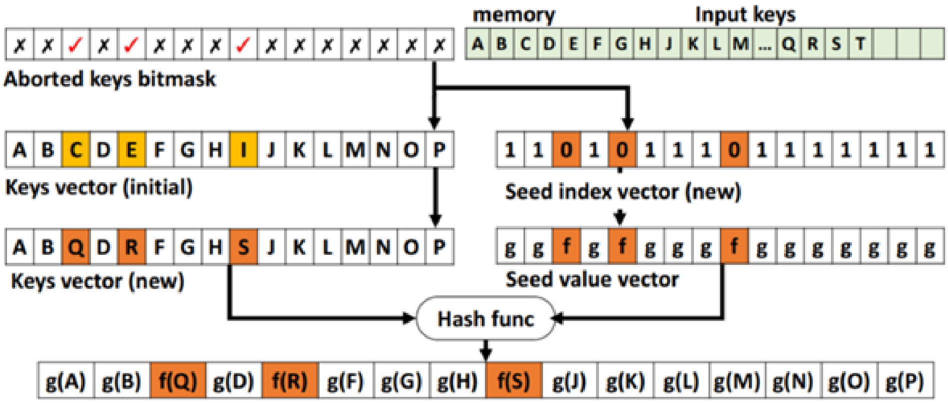

memory
Input keys
A B C D E F G H J K L M ... Q R S T
✗ ✗ ✓ ✗ ✓ ✗ ✗ ✗ ✓ ✗ ✗ ✗ ✗ ✗ ✗ ✗
Aborted keys bitmask
A B C D E F G H I J K L M N O P
Keys vector (initial)
1 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1
Seed index vector (new)
A B Q D R F G H S J K L M N O P
Keys vector (new)
g g f g f g g g f g g g g g g g
Seed value vector
Hash func
g(A) g(B) f(Q) g(D) f(R) g(F) g(G) g(H) f(S) g(J) g(K) g(L) g(M) g(N) g(O) g(P)

Figure 1.10: Vectorized Bloom Filter [13]

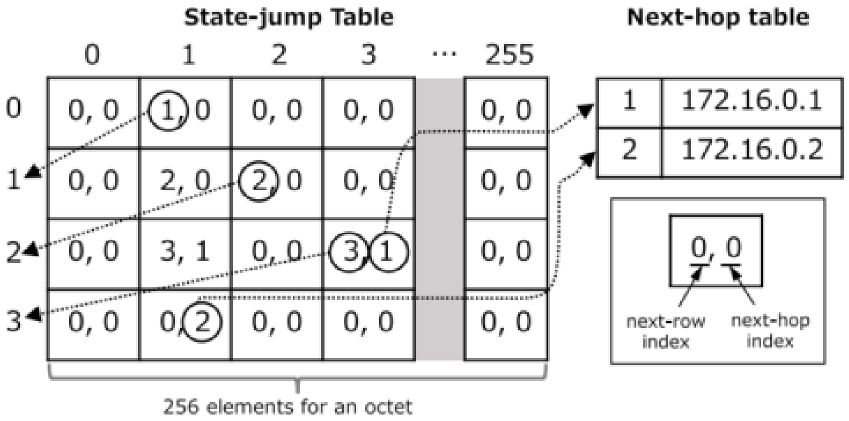
To further increase the performance, the authors implement multi-way loop unrolling. Loop unrolling itself lets the processor use out-of-order execution to increase the parallelization of the code. The authors use multi-way loop unrolling by partitioning the input array into different key vectors. In each iteration of the loop, a part of those vectors is processed in parallel until one of them is finished. The rest is processed sequentially. Additionally, if the bloom filter exceeds the cache size, then the performance degrades because different words of the filter have to be fetched and cache misses occur. To combat this, they employ memory access latency hiding by prefetching parts of the bloom filter [13].

The authors call the combination of those techniques highly vectorized bloom filter (HPBF) and were able to show in practical evaluation that the performance of the membership check is significant and can reach up to 162% in certain settings. Nevertheless, these results have to be taken with a grain of salt, since naively using SIMD parallelism did not show any improvement in performance [13]. If we compare the HPBF to the UFBF in detail, we see that even though the authors had the same kind of approach, they employed different strategies to be able to fully capitalize on the SIMD processor parallelism.

1.3.2 Longest Prefix Matching

When routers route internet traffic, they typically look up the IP destination address of the packet/datagram in their routing table to know where to forward it. The entries of the routing table are made up of an IP address and a corresponding subnet mask. A destination address can now match with two entries with different length subnet masks at the same time. To break ties, the whole address of both of the entries is looked at and the one with the longest matching prefix is chosen [4]. Ueno et al. [22] identify a need for software middleboxes to be able to perform longest prefix matching (LPM) to handle multiple 100 Gbit/s interfaces. They propose an LPM method called Spider, which utilizes SIMD processing to gain a speed advantage.

To be able to benefit from SIMD processing, the authors avoid pointer referencing and bitwise operations during the lookup procedure. The SIMD operations are unfit to efficiently parallelize pointer referencing. To avoid this, Ueno et al. [22] use a state-jump table (see Figure 1.11).



	0	1	2	3	···	255
0	0, 0	1, 0	0, 0	0, 0		0, 0
1	0, 0	2, 0	2, 0	0, 0		0, 0
2	0, 0	3, 1	0, 0	3, 1		0, 0
3	0, 0	0, 2	0, 0	0, 0		0, 0

1	172.16.0.1
2	172.16.0.2

Figure 1.11: State-Jump Table [22]

Using an appropriate fixed stride length and aligning the data bytewise enables to efficiently use SIMD instructions for parallel processing. For the first two octets direct lookup table replaces the first two rows of the state-jump table. This is because for two octets, the table can provide a significant speedup while still maintaining a tolerable size. To avoid pointer referencing of the direct lookup table to the appropriate row of the state-jump table, they are both located next to each other in the memory (see Figure 1.12).

The IPv4 version of the algorithm (see Figure 1.21) can now do the lookup procedure for 8 destination addresses given eight 32-bit operations are supported in parallel. The first two octets are extracted from the destination address and the corresponding index of their next hop index (NHI) and next row index (NRI) is calculated. At this index, the value representing the NHI and the NRI is located and both of them are extracted. The NHI is the current result. Now, for the rest of the rows of the state-jump table, the next octet is extracted. Using the current NRI again, the index of the next NHI and NRI are calculated, and they are extracted. Using the blending operation, we determine the current NHI. After our NRI becomes zero, indicating the end of the state-jump table, it stops and stores the result [22]. Because of the memory layout of the data structure, all these steps can be realized using SIMD operations.

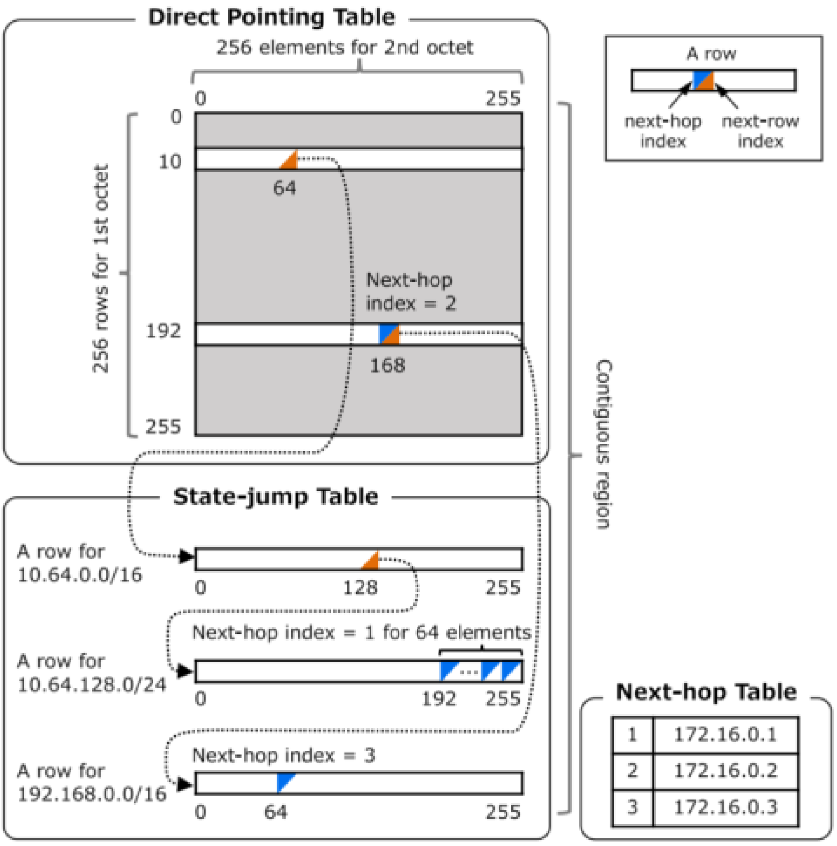


Figure 1.12: Spider Data Structure [22]

The state-jump table Lu et al. [22] propose does not support updating operations in the routing table. To circumvent this, an alternative representation of the routing table using a multiway trie is used. The authors employ a batch-based update system of the routing table where after a certain time frame the routes of the multiway trie which keeps track of the updates are aggregated and converted into the direct lookup and state-jump table. This allows for regular updates without performance degradation.

For simplicity, the IPv4 version of Spider was presented, but Lu et al. [22] show that it can be augmented to serve IPv6 addresses with minor changes. Spider was compared to state-of-the-art methods for LPM in software middleboxes (PopTrie and DXR). The creators of Spider were able to show that for high-frequency processors the method is 1.8-3.2 times faster (see Figure 1.13) and theoretically enables to serve up to 34 Gbit/s interfaces at the same time.

It is also the case with LPM that dull application of SIMD processing cannot yield a better performance. Specialized data structures need to be used to enable the utilization of parallelizable SIMD operations. Furthermore, these data structures can impose new limitations which need to be addressed with innovative solutions. By overcoming these challenges, the creators of Spider showed that SIMD processing for software networking applications yields a performance improvement.

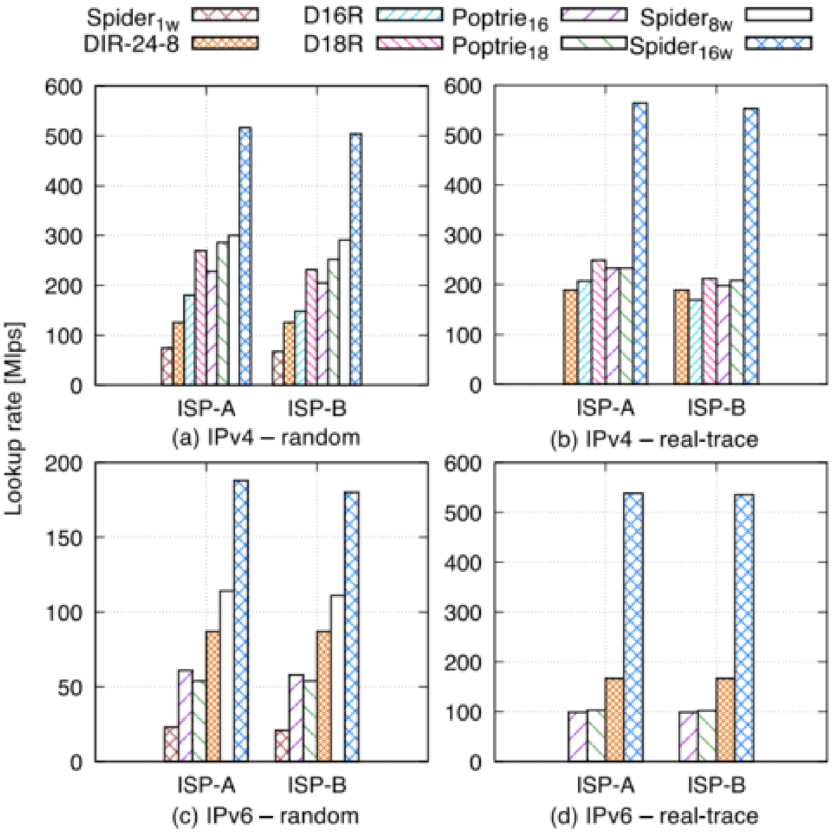


Figure 1.13: Spider Comparison [22]

1.3.3 Deep Packet Inspectiont

Han et al. [7] identify a lack of network trafficking monitors that support in-depth analysis at high speed and provide accurate timestamping. The authors implement a measurement system consisting of a field programmable gateway (FPGA) and data plane development kit (DPDK) [17] technology for fast data packet preprocessing on the host system. The FPGA uses direct memory access (DMA) to copy data from its memory to the host memory. SIMD processing is employed to transfer data from the DMA cache to the memory buffer (see Figure 1.14). Compared to the conventional copy operations, the SIMD instruction set enables a performance boost from 12 to 17 Gbit/s throughput.

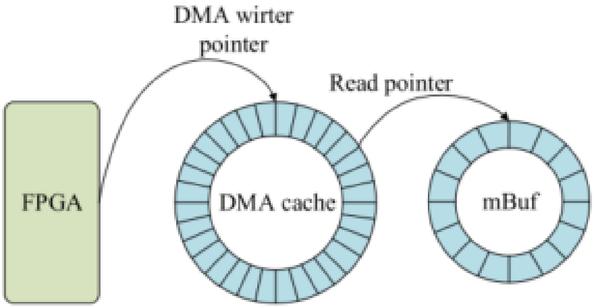


Figure 1.14: Direct Memory Access [7]

For deep packet inspection, the multi-pattern regex matcher Hyperscan [24] is used. Within Hyperscan, Harry [25], a multi-literal pattern matching engine pre-filters the inputs before the actual regex matching. Most DPI applications use the standard Aho Corasick (AC) algorithm [1], which is much slower than its SIMD processing alternatives. Before Harry was integrated within Hyperscan FDR an engine already able to do SIMD processing was employed but was still the bottleneck of the application. In the following, it is explained how the standard Shift-Or Algorithm was changed to use SIMD processing and what Xu et al. did to further enhance SIMD utilization.

FDR as well as Harry are based on the Shift-Or algorithm. Figure 1.15 shows an example of the single-literal Shift-Or algorithm, where the literal ’rry’ is matched on the input string ’rsyrry’. In the mask table (see Figure 1.15 (A)) the entries are set to zero if the character of the literal on top matches the corresponding input string on the side and one if there is no match. In the first iteration, the entries of the first two characters of the input string are loaded into the match table. Colored diagonal entries in Figure 1.15 (B) correspond to trying to match the literal onto the input string. The green diagonal entries for example correspond to matching ’rry’ onto the part ’rsy’ of the input string. To parallelize this matching, the match table is shifted as seen in Figure 1.15 (C) and an or operation is used on the entries for a matching attempt. This results in the state mask of the first iteration (Figure 1.15 (D)). Steps (A) to (D) are repeated for the part ’yrry’ of the input string and again the or operation is used on both state masks. If in the end, a zero can be read in the updated state mask, then a match took place at that position. In the example, a zero can be read on index three, meaning that the literal ’rry’ matches the input string ’rsyrry’ from index three on [25].

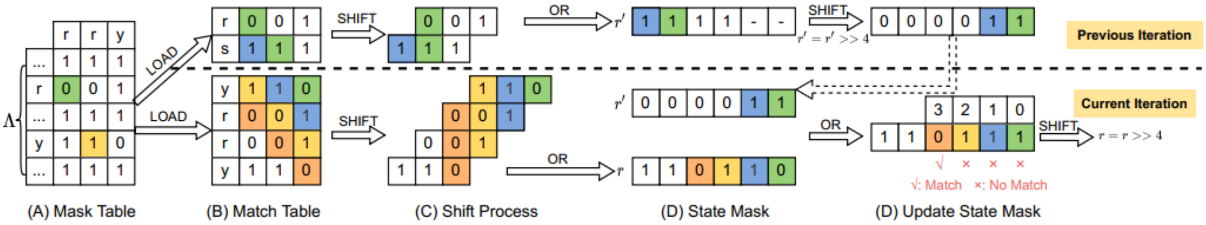


Figure 1.15: Single-Literal Shift-Or Matching [25]

To move from single-literal to multi-literal Shift-Or several literals are included in the mask table. The width of the mask table now becomes the width of the longest literal and the bits in the entries of the mask table are set to zero if the character of the corresponding literal matches the character of the input string (see Figure 1.16) [25].

The first SIMD-enabled multi-literal Shift-Or matching that FDR employs is based on the row-vector Shift-Or model (see Figure 1.17). The mask table has a width of $n \cdot l$ where n is the number of literals and l is the maximal literal length. m rows are processed in one iteration. These are loaded into the match table and are shifted such that we end up with a shifted table of length $n(l + m - 1)$ bits. If the SIMD vector length is L then $n(l+m-1) < L$ has to hold. FDR chooses n and l to be eight. If n has to be larger, FDR will employ a mechanism called grouping which buckets together literals and matches the buckets with reduced accuracy. If l is larger, FDR uses truncation of the literals which also introduces more false positives [25].

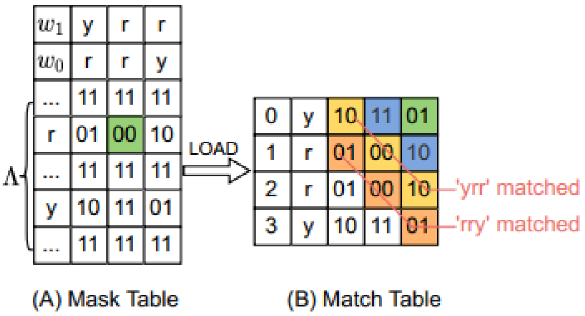


Figure 1.16: Multi-Literal Shift-Or Matching [25]

The problem with FDR is that even though a wider SIMD vector allows for a larger m. FDR still needs $3m$ SIMD operations (LOAD, SHIFT, and OR) to deal with those characters. So due to poor data-level parallelism FDR does not profit much from wider SIMD vector instructions. Further, FDR has a fixed 64-bit mask and therefore a low SIMD utilization. FDR can also not be implemented in AVX512 without major shortcomings because of the implementation of the shift instruction [25].

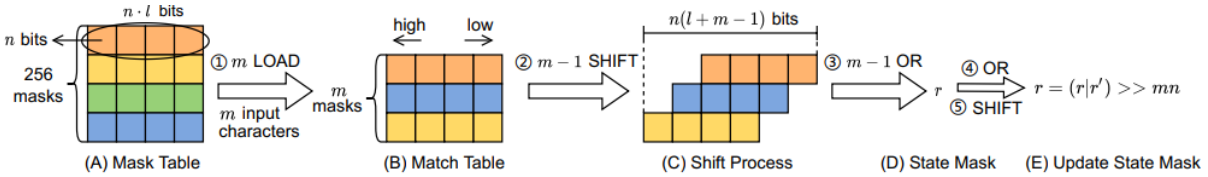


Figure 1.17: Row-Vector-Based Shift-Or Model [25]

To combat these shortcomings, Harry is based on the column-vector Shift-Or model (see Figure 1.18) and uses a shuffle instead of a shift instruction. The basics of the algorithm are the same, except that the data layout changes. The column-vector-based model uses the fact that with the alternate data layout, a vertical shift works as well. The upper bound for SIMD operations needed per iteration is $3l$ and independent of m. Given we choose n and l to be eight and the AVX512 instructions are used, m can be 56 without exceeding the 512-bit vector. To be able to choose n and l to be eight, Harry uses the same strategies, grouping and truncation. In this optimal case, only 0.41 SIMD operations per input character are needed, and the vector utilization is 87.5% instead of 12.5% for FDR [25].

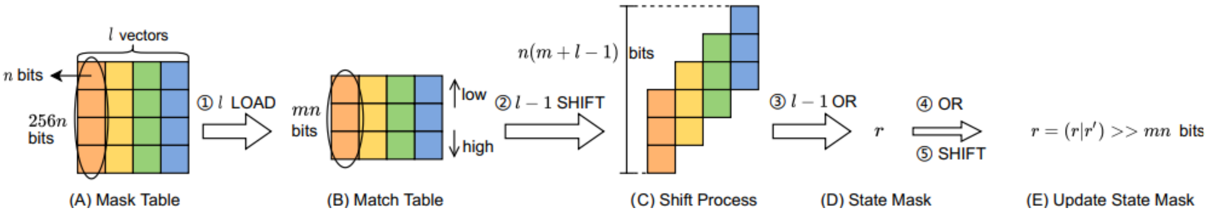
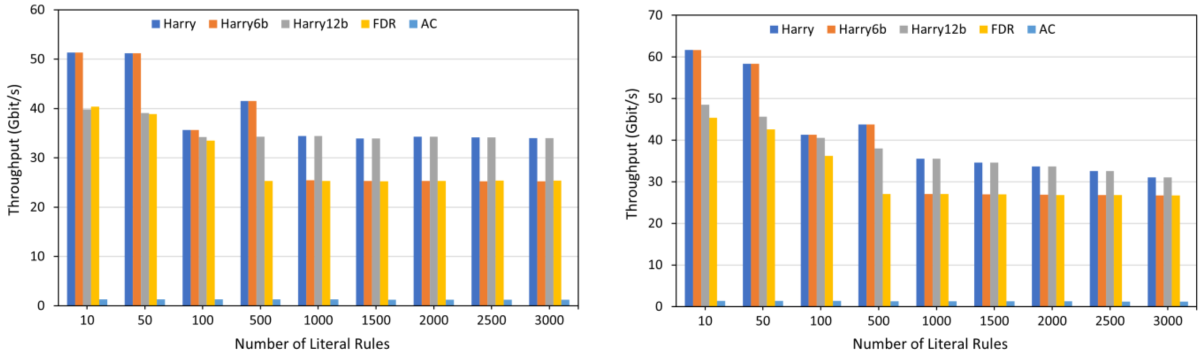


Figure 1.18: Column-Vector-Based Shift-Or Model [25]

Since we chose n to be eight, the SIMD vector would need to have a 2048-bit width. Naively, using this algorithm is impossible at the moment because the longest available

SIMD instructions are still 512-bit long. To circumvent this problem, two encoding mechanisms are used that reduce the 2048-bit long vectors to 512-bit. A stronger encoding model Harry6b loses more information but does not require additional SIMD operations. Harry12b loses less information but requires additional SIMD operations. It happens that the loss of information for a large number of literals introduces so many false positives that the cost of correction operations overtakes the benefit. Figure 1.19 shows the results of the evaluation for literals of the OWASP ModSecurity core rule set and HTTP as well as non-HTTP packets [25]. The higher the number of literal rules, the smaller the performance difference to FDR. Their evaluations show that for the Snort community rule set and the OWASP ModSecurity core rule set the performance is similar to FDR for 3000 literals. Both rule sets have more than 3500 literal if the whole rule set is used. It is assumed that the true strength of Harry lies in its scalability.



(a) IXIA HTTP Packets

(b) Alexa Non-HTTP Packets

Figure 1.19: ModSecurity Ruleset Comparison of Harry [25]

1.3.4 Random Network Coding

Random Linear Network Coding (RLNC) is a technique that is used to reliably distribute packages in a network. In a traditional network, packages are routed through a predefined path. However, RLNC enables us to combine different packages at multiple intermediate nodes into an encoded form. This allows a package to contain more information in one package about the source data than in other methods. RLNC has therefore a high chance to deliver the data to all the end nodes successfully and is resistant to package loss. The use of large enough finite fields for coding is the success factor for RLNC and makes the failure rate of delivering a package very small [10].

A concrete example of RLNC would be that we have a network of multiple nodes, and we want to send packages from a source node S to a sink node K over multiple nodes. We can now choose random coefficients from a Galois field and create with the random coefficients a random linear combination of the packets we want to send. If the sink node S has a sufficient linear independent combination of packages, the node can decode them using Gaussian elimination. With this technique, S can restore the original packages. This also works when intermediate nodes generate new linear combinations of the packages [9].

Sørensen et al. [20] examined in their research paper the efficiency of RLNC regarding energy. RLNC is an efficient coding method, however it also requires a lot of energy. This is especially the case for battery-powered devices like mobile phones and sensor devices. Therefore, they wanted to study the energy cost of RLNC on different platforms and wanted to find if SIMD operations can make RLNC more energy efficient. With SIMD they could encode packages faster since they could do 16 multiplications simultaneously. The results of their experiment showed that making use of SIMD hardware optimizations benefits energy usage as well as processing speed. Results show that SIMD operations provide a speedup of 4 to 18 times. This makes mobile devices using SIMD hardware comparably fast, as Intel i5 processors without SIMD.

A paper by Shin et al. [18] focused on using different SIMD extensions for a speedup in RLNC. They focused on architectures with SIMD register sizes. Therefore, they studied AVX (128-bit), AVX2 (256-bit), and AVX512 (512-bit) SIMD extensions. To use RLNC, Galois field arithmetic is fundamental, which was the focus of this paper. Results show that the AVX512 extension outperforms the other extensions significantly. AVX512 had a 58% higher throughput than AVX and a 26% higher throughput than AVX2. Therefore, the paper demonstrates the efficiency gain of using SIMD operations for RLNC.

1.3.5 IoT Authentication

The number of Internet of Things (IoT) devices in use is growing, and more and more people are making use of the new technology. However, with the growing number of IoT devices, malware targeting those devices has also become more popular. That is why Choi et al. [3] propose a high-speed and furthermore lightweight authentication protocol that also runs on low-powered IoT devices. Exactly due to the limited power of the devices, the authors tried to use technologies that compensate for this. Therefore, they make use of SIMD extensions. Since IoT devices often use ARM processors, they designed their system for the NEON SIMD extension. Their protocol additionally makes use of two more technologies. For encryption, the protocol uses LEA-128-CTR, which is a lightweight encryption algorithm and for integrity checking the protocol requires Chaskey MAC algorithm which is especially suited for 32-bit microcontrollers.

Size of Auth_Data	**Processing Time (ms)**		**Speed ratio of SIMD (A/B)**
	Non-SIMD (A)	**SIMD (B)**	
1k	2.364	1.862	1.27
10k	2.403	1.914	1.26
100k	2.753	2.036	1.35
1M	6.379	4.396	1.45
2M	10.364	6.397	1.62
5M	22.391	12.764	1.75
10M	42.354	22.770	1.86

Table 1.2: Processing time data of Non-SIMD and SIMD [3]

In their evaluation, they experimented with different authentication data sizes. As it can be seen in Table 1.2 was the processing always faster when they made use of the SIMD capabilities of the devices. Additionally, it can be seen that with growing authentication data, the processing time is relatively faster. For the size of 1000, it was a speedup of 21% while for a size of 10M, it was 46%.

1.3.6 IoT Security

IoT devices often use lightweight cryptography to protect their data. However, one problem that IoT devices can be affected by is fault attacks. Fault attacks often include an attack on the hardware of the device to introduce faults to the cryptographic processes. Preventing those attacks is often costly and also difficult for off-the-shelf devices. To solve this problem, Lac et al. [12] propose in their paper a method to prevent those fault attacks by using an Internal Redundancy Countermeasure (IRC) which makes use of SIMD instructions. The principle of IRC is that instead of making full use of the 32-bit architecture, and computing 32-bit ciphers, the 32 bits are divided into 4 blocks of 8-bit words, which serves as spatial redundancy. Those data blocks consist of data as well as reference values. Data blocks are always separated with reference values blocks, which serve as an additional control mechanism.

There exist two possibilities to make use of using SIMD on block ciphers, one is called fault detection and the other is called fault correction. Fault detection uses multiple copies of the data. If the outcome of the multiple data blocks does not lead to the same result, the system is trapped. Fault correction however also computes multiple copies of the data concurrently. However, after the computation, there is a majority vote. The computation that wins the majority vote is then returned.

Since their focus was on IoT devices, the IRC was tested on ARM Cortex-M3 and Cortex-M4 processors. To conduct the experiments, they made use of the block cipher algorithm PRIDE and the stream cipher TRIVIUM. The results showed that IRC needs a higher cycle count as well as more memory, however, they could show that fault attacks could be successfully thwarted. They conclude that utilizing SIMD instructions for spatial redundancy effectively enhances fault resistance in lightweight cryptography. The method is a trade-off between performance and security, however, it is suitable for a wide variety of IoT devices since it does not require any hardware modification.

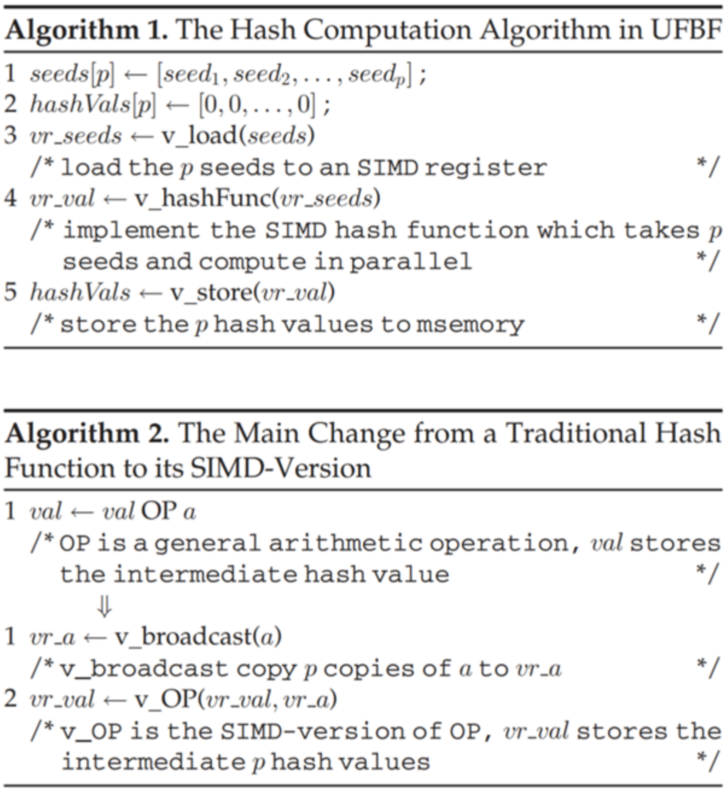
1.4 Summary and Conclusions

All presented papers show that SIMD processing requires sophisticated data layouts to make use of parallel computation. This is because of the limited size of the SIMD registers or vectors and the time it takes to load the data into the SIMD registers before being able to perform operations on it. This should have become evident to the reader during the in-depth presentation of the bloom filter [14] [13], longest prefix matching [22], and deep packet inspection [25] applications. The register/vector size often forces the developers to accept trade-offs where their architecture allows better performance on big registers but in the real world they have to settle for smaller ones which curbs the performance. In the case of the ultra-fast bloom filters to be able to use more hash functions, a drop in performance was accepted [14]. In Harry, the pattern matching engine encoding schemes were used to fit the data into the registers, which resulted in lower throughput [25]. When using SIMD for the longest prefix matching the size of the registers was not a problem, but the fact that the data structure was needed made it impossible to dynamically change the routing table. This was used by introducing a separate management data structure to overcome this problem [22].

In two papers, it also became apparent that using twice the register/vector size does not guarantee twice the performance. The use of 512-bit vectors in random linear network coding only enhanced the throughput by 58% compared to 128-bit and 26% compared to 256-bit [18]. In the case of deep packet inspection when applying Harry (512-bit) in real-world scenarios, the performance gain was only factors of 1.06-1.63 and 1.14-1.62 compared to FDR (128-bit). Even if Harry had a much higher character processing per SIMD instruction [25]. This raises the question of whether it is even useful to build larger and larger SIMD registers and vectors. It seems that algorithms like Harry can profit from it at some point. For example, a 2048-bit register would prevent the use of encoding for 256 masks [25] but the increase in size can not be translated easily into the same increase in performance. Maybe at some point, the cost of changing computer architecture such that the SIMD vectors are even wider outweighs the benefits.

The results were mixed for the bloom filter applications. On one hand, the vectorized bloom filters [13] showed promising results. On the other hand, the trade-off that has to be made for the ultra-fast bloom filter [14] raises the question of whether the real performance gain for real-world application is this significant. A similar problem arises with Harry [25]. Even though Harry is integrated in Hyperscan [24] instead of FDR the results in real-world settings seem not convincing that it was worth the effort to develop an upgrade from FDR to Harry. It must be noted that these are opinions of the authors of this overview and have to be taken with a grain of salt since they are not experts in the field of SIMD application. Another trade-off had to be made in the SIMD application in IoT security. The authors were able to thwart fault attacks, but the overall performance of the system went down [12]. The results of IoT authentication were only up to 46%, but no trade-off was reported. Good results yielded the SIMD processing in longest prefix matching [22] and random linear network coding in mobile devices [20]. Since the performance increases were good and no trade-offs were reported either. It is concluded that the use of SIMD processing in network applications is all but trivial. Nevertheless, network architects should consider it when a performance boost is needed.

1.5 Appendix A: Algorithms



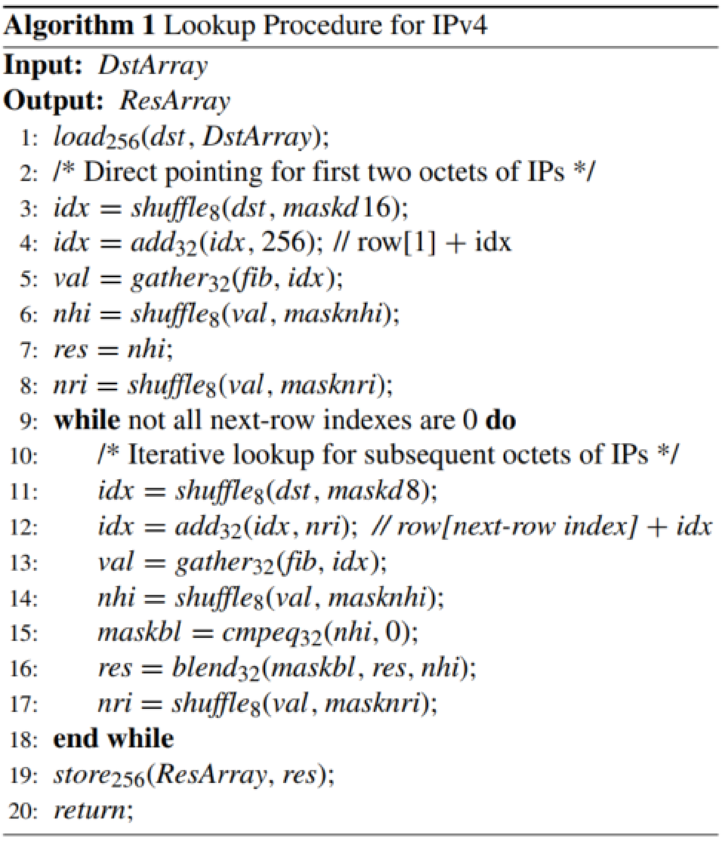
Algorithm 1. The Hash Computation Algorithm in UFBF

```
$seeds[p] \leftarrow [seed_1, seed_2, \ldots, seed_p]$ ;
$hashVals[p] \leftarrow [0, 0, \ldots, 0]$ ;
$vr\_seeds \leftarrow$ v_load($seeds$)
/* load the $p$ seeds to an SIMD register */
$vr\_val \leftarrow$ v_hashFunc($vr\_seeds$)
/* implement the SIMD hash function which takes $p$
   seeds and compute in parallel */
$hashVals \leftarrow$ v_store($vr\_val$)
/* store the $p$ hash values to msemory */
```

Algorithm 2. The Main Change from a Traditional Hash Function to its SIMD-Version

```
$val \leftarrow val$ OP $a$
/* OP is a general arithmetic operation, $val$ stores
   the intermediate hash value */
     $\Downarrow$
$vr\_a \leftarrow$ v_broadcast($a$)
/* v_broadcast copy $p$ copies of $a$ to $vr\_a$ */
$vr\_val \leftarrow$ v_OP($vr\_val, vr\_a$)
/* v_OP is the SIMD-version of OP, $vr\_val$ stores the
   intermediate $p$ hash values */
```

Figure 1.20: UFBF Parallelization [14]



```
Algorithm 1 Lookup Procedure for IPv4
Input: DstArray
Output: ResArray
$load_{256}(dst, DstArray)$;
/* Direct pointing for first two octets of IPs */
$idx = shuffle_8(dst, maskd16)$;
$idx = add_{32}(idx, 256)$; // row[1] + idx
$val = gather_{32}(fib, idx)$;
$nhi = shuffle_8(val, masknhi)$;
$res = nhi$;
$nri = shuffle_8(val, masknri)$;
while not all next-row indexes are 0 do
    /* Iterative lookup for subsequent octets of IPs */
    $idx = shuffle_8(dst, maskd8)$;
    $idx = add_{32}(idx, nri)$; // row[next-row index] + idx
    $val = gather_{32}(fib, idx)$;
    $nhi = shuffle_8(val, masknhi)$;
    $maskbl = cmpeq_{32}(nhi, 0)$;
    $res = blend_{32}(maskbl, res, nhi)$;
    $nri = shuffle_8(val, masknri)$;
end while
$store_{256}(ResArray, res)$;
$return$;
```

Figure 1.21: Spider Lookup Procedure [22]

Bibliography

[1] Alfred V. Aho and Margaret J. Corasick. "Efficient string matching: an aid to bibliographic search". In: 18.6 (June 1975), pp. 333–340. DOI: `10.1145/360825.360855`. URL: `https://doi.org/10.1145/360825.360855`.

[2] Andrei Broder and Michael Mitzenmacher. "Survey: Network Applications of Bloom Filters: A Survey." In: *Internet Mathematics* 1 (Nov. 2003). DOI: `10.1080/15427951.2004.10129096`.

[3] Seul-Ki Choi, Ju-Seong Ko, and Jin Kwak. "A study on IoT device authentication protocol for high speed and lightweight". In: *2019 international conference on platform technology and service (PlatCon)*. IEEE. 2019, pp. 1–5.

[4] Douglas Comer. *Computer Networks and Internets*. Pearson/Prentice Hall, 2009. ISBN: 9780136061274. URL: `https://books.google.ch/books?id=tm-evHmOs3oC`.

[5] Michael J Flynn. "Very high-speed computing systems". In: *Proceedings of the IEEE* 54.12 (1966), pp. 1901–1909.

[6] Gartner. *Public cloud application services/software as a service (SaaS) end-user spending worldwide from 2015 to 2024 (in billion U.S. dollars) [Graph]*. `https://www.statista.com/statistics/505243/worldwide-software-as-a-service-revenue/`. 2023.

[7] Luchao Han et al. "A Multifunctional Full-Packet Capture and Network Measurement System Supporting Nanosecond Timestamp and Real-Time Analysis". In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–12. DOI: `10.1109/TIM.2021.3080375`.

[8] M. Hassaballah, Saleh Omran, and Youssef B. Mahdy. "A Review of SIMD Multimedia Extensions and their Usage in Scientific and Engineering Applications". In: *The Computer Journal* 51.6 (Jan. 2008), pp. 630–649. ISSN: 0010-4620. DOI: `10.1093/comjnl/bxm099`. URL: `https://doi.org/10.1093/comjnl/bxm099`.

[9] Tracey Ho and Desmond Lun. *Network coding: an introduction.* Cambridge University Press, 2008.

[10] Tracey Ho et al. "The benefits of coding over routing in a randomized setting". In: *IEEE international symposium on information theory.* 2003, pp. 442–442.

[11] Nicklas Bo Jensen and Sven Karlsson. "Improving loop dependence analysis". In: *ACM Transactions on Architecture and Code Optimization (TACO)* 14.3 (2017), pp. 1–24.

[12] Benjamin Lac et al. "Thwarting fault attacks against lightweight cryptography using SIMD instructions". In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2018, pp. 1–5.

[13] Hejing Li, Juhyeng Han, and Dongsu Han. "Leveraging SIMD parallelism for accelerating network applications". In: *Proceedings of the 4th Asia-Pacific Workshop on*

Networking. Seoul, Republic of Korea, 2020, pp. 23–29. DOI: 10.1145/3411029.3411033.

[14] Jianyuan Lu et al. "Ultra-Fast Bloom Filters using SIMD techniques". In: *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*. 2017, pp. 1–6. DOI: 10.1109/IWQoS.2017.7969125.

[15] Onur Mutlu. *Lecture 25: SIMD Processors and GPUs*. Lecture Slides. 2024-05-19. Jan. 2023. URL: https://safari.ethz.ch/architecture/fall2022/lib/exe/fetch.php?media=onur-comparch-fall2022-lecture25-simd-processors-and-gpu-beforelecture.pdf.

[16] Alex Peleg and Uri Weiser. "MMX technology extension to the Intel architecture". In: *IEEE micro* 16.4 (1996), pp. 42–50.

[17] DPDK Project. LF Projects. *DPDK*. Available Online. 2024-05-19. URL: https://www.cs.umd.edu/users/meesh/cmsc411/website/projects/SIMDproj/project.html.

[18] Seo-Ran Shin, Se-Yeon Choo, and Joon-Sang Park. "Accelerating Random Network Coding using 512-bit SIMD Instructions". In: *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. 2019, pp. 1099–1103.

[19] Keith Slutskin and Kasima Tharpipitchai. *Into The Fray With SIMD*. Available Online. 2024-05-19. URL: https://www.cs.umd.edu/users/meesh/cmsc411/website/projects/SIMDproj/project.html.

[20] Chres W Sørensen et al. "Leaner and meaner: Network coding in SIMD enabled commercial devices". In: *2016 IEEE Wireless Communications and Networking Conference*. IEEE. 2016, pp. 1–6.

[21] Alfred Spector and David Gifford. "The space shuttle primary computer system". In: *Communications of the ACM* 27.9 (1984), pp. 872–900.

[22] Yukito Ueno et al. "Fast Longest Prefix Matching by Exploiting SIMD Instructions". In: *IEEE Access* 8 (2020), pp. 167027–167041. DOI: 10.1109/ACCESS.2020.3023156.

[23] Yukito Ueno et al. "Spider: Parallelizing Longest Prefix Matching with Optimization for SIMD Instructions". In: *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. 2020, pp. 267–271. DOI: 10.1109/NetSoft48620.2020.9165306.

[24] Xiang Wang et al. "Hyperscan: A Fast Multi-pattern Regex Matcher for Modern CPUs". In: *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. Boston, MA: USENIX Association, 2019, pp. 631–648. ISBN: 978-1-931971-49-2. URL: https://www.usenix.org/conference/nsdi19/presentation/wang-xiang.

[25] Hao Xu et al. "Harry: A Scalable SIMD-based Multi-literal Pattern Matching Engine for Deep Packet Inspection". In: *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*. 2023, pp. 1–10. DOI: 10.1109/INFOCOM53939.2023.10229022.

[26] Jingren Zhou and Kenneth A Ross. "Implementing database operations using SIMD instructions". In: *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*. 2002, pp. 145–156.

Chapter 2

Privacy Preserving Synthetic Data Generation: A Taxonomy and Scoping Review

Joshua Stebler, Faye Dinh

The proliferation of big data has led to rapid advancements in machine learning, but it has also raised concerns about data privacy. Machine learning algorithms can reverse blurred images or uncover relationships in data that may lead to the deanonymisation of private data. In this paper, we discuss how synthetic data can be used to protect personal data by not sharing the original data but instead sharing new data that follows the same statistical properties as the real data. We will introduce various synthetic data generation frameworks and explain the models typically used for each data modality. Also we will discuss the limitations and challenges within the taxonomy of synthetic data.

Contents

2.1 Introduction

Due to the rapid advancements in data-driven technologies, the capabilities of artificial intelligence continue to expand. However, these advancements also raise concerns about data privacy. Machine learning is getting better at deblurring images [31] and showing complex relations in datasets that can lead to the association of anonymised data to personal information. Synthetic data generation offers a solution by allowing the use of realistic datasets while ensuring the privacy of individuals involved. This report presents a taxonomy of methods for privacy-preserving synthetic data generation.Synthetic data generation involves creating new datasets that are statistically indistinguishable from real datasets but contain no actual data points from the original dataset. This preserves the privacy of the individuals in the original dataset, allowing the synthetic data to be shared and utilized without compromising privacy [38].

This report examines the models used to generate various data types-image, text, tabular, time series, location, and genomics data-and discusses the specific challenges associated with each. We explore the application of differential privacy [24] techniques to Generative Adversarial Networks(GAN) [18], Variational Autoencoders(VAE) [31] and other machine learning models. These methods are prominently used for synthetic data generation. The aim of this report is to provide a comprehensive overview of synthetic data generation models, detailing how synthetic data is generated and which models are typically used for various data modalities. By explaining the methodologies and challenges of synthetic data generation, this report wants to provide an understanding of this topic and spark interest, hopefully leading to further research.

2.2 Background

In this report, we will discuss how to create privacy-preserving synthetic data. It should be possible to share this data and work with it without compromising the privacy of the data set used to train the model. We will then explore the models used for different data modalities and point out difficulties.

2.2.1 Synthetic Data

Many conventional anonymisation techniques are becoming increasingly insecure, especially since the advancements in machine learning that have become very good at denoising a picture or detecting patterns in data that can be used to assign the data to a person. This is where synthetic data comes into play; it tries to avoid this problem not by better anonymising the data, but by creating new data that follows the same statistical properties of the original data. This way, you have realistic data you can work with but has never existed before.

2.2.2 Gradient Descent Algorithm

Gradient Descent(SGD) [42] is a frequently used stochastic optimisation method employed in machine learning training. It derives the Mean Squared Error (MSE) over the whole loss function, and from which the direction of the steepest gradient is derived. Changing the parameters in that direction then minimises the MSE the fastest. Batch Gradient Descent is guaranteed to converge to a global minimum for a convex function and to a local minimum for a non-convex function. Over multiple iterations, the algorithm approaches the minimum of the loss function stepwise, with a stepsize called ”learning rate”. This is a hyperparameter in training settings. If the chosen learning rate is too large, the algorithm might miss the global minimum, and if it’s too small, we might get stuck in a local minimum rather than the global minimum. To make this algorithm less computationally demanding, stochastic gradient descent can be used, which does not always compute the whole gradient. This can allow it to skip a non-optimal local minima but also makes it possible to overshoot the global minimum; however, it has been shown to be very effective [4]. Notable for *differentially private* machine learning is the modification DP-SGD, by Abadi et al [3]. This paper introduces the gradient clipping and noising techniques and a privacy accountant called Moments accountant to track the privacy loss incurred during training. Gradient Clipping ensures that no individual sample has too much influence on the gradient, as that could compromise safety. The clipped gradients $\overline{\mathbf{g}}_t(x_i) = \mathbf{g}_t(x_i)/\max(1, \frac{\|g_t(x^{(i)})\|}{C})$, where $\mathbf{g}_t(x_i)$ are the gradients calculated from L samples, the lot, C is the clipping threshold and t is the current training step the algorithm is in. L, C are supplied as hyperparameters. This ensures that the gradients where $\|\mathbf{g}\|_2 \geq C$ are scaled down, which reduced the influence of outlier samples. Further noise sampled from a multivariate gaussian distribution $\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$ is sampled and added onto each gradient to obscure which lot it was calculated from. The choice of σ is made dependent on ε and δ (which are explicit parameters given and quantify (ε, δ)-privacy).

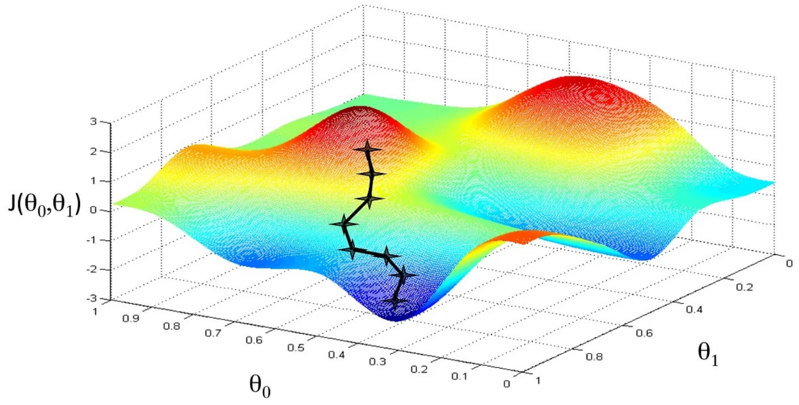


Figure 2.1: This visualisation shows a cost function in machine learning where θ_0 and θ_1 represent parameters that can be optimized, the error of the of the machine learning model $J(\theta_0, \theta_1)$ is also represented as a colour.[4]

2.2.3 GAN

We provide a summary of Generative Adversarial Networks(GAN), as proposed by Ian Goodfellow et al. in [18]:
Generative Adversarial Networks consist of a generator neural network G and a discriminator or critic neural network D. The generator G is given some input $\mathbf{z}$ from a prior noise distribution p_z, and represents some differentiable function with parameters θ_G which maps $\mathbf{z}$ to a posterior distribution p_G. The discriminator network receives an input $\mathbf{x}$ and returns the probability that $\mathbf{x} \sim p_{real}$ rather than $\mathbf{x} \sim p_G$. The training goal for D is then to maximise the accuracy of classification, while G's training goal is to simultaneously minimise the probability that its output is classified as not being sampled from p_{real}, quantified as $\log(1 - D(G(\mathbf{z})))$. This can be formalised as G and D playing a minimax game with the condition

$$\min_G \max_D \mathbb{E}\left[\log\left(D(\mathbf{x})\right] + \mathbb{E}\left[\log\left(1 - D(G(\mathbf{z})\right)\right] \;\; \forall \mathbf{x} \sim p_{real}, \forall \mathbf{z} \sim p_G$$

. The training is complete when $p_G = p_{real}$ and $D(\mathbf{x}) = \frac{1}{2}$ The authors note that in early training phases $\log(1 - D(G(\mathbf{z})))$ may dominate, because generated samples are not very close to the real distribution yet. Instead they propose that G should learn to maximise $log(D(G(\mathbf{z}))$.

2.2.4 VAE

The Stochastic Gradient Variational Bayes estimator(SGVB) proposed by Kingma and Welling(2022)[31], when applied to auto-encoder neural networks yields the variational auto-encoder(VAE) that is in prevalent use for a multitude of generation tasks, as we will also demonstrate in the report. Like the autoencoder, it consists of an encoder and a decoder, which Kingma and Welling implement as Multilayer Perceptrons. In Autoencoders the encoder network approximates a posterior probability distribution $p_\phi(\mathbf{z}|\mathbf{x})$ of a latent variable or code $\mathbf{z}$ given some observed data point $\mathbf{x}$. Conversely the decoder network approximates the posterior probability $p_\theta(\mathbf{x}|\mathbf{z})$, the probability distribution of the value of $\mathbf{x}$ given the latent code $\mathbf{z}$. The challenge however is that these posteriors can be intractable, and therefore not differentiable. This is where the authors apply their proposed SGVB estimator to the intractable posteriors, to derive a differentiable lower bound, and thus make the application of a statistical optimisation algorithm to the posteriors possible. The trick the authors apply is to reparameterise the latent code $\mathbf{z}$ from a sample of an intractable distribution to a deterministic

$$\mathbf{z} = g_\phi(\epsilon, \mathbf{x}),\text{where } \epsilon \sim p(\epsilon)$$

. This function g_ϕ is a vector-valued function, parameterised by ϕ and must map the data point and ϵ to $\mathbf{z}$. The distribution over ϵ, crucially, is independent of the posterior, and therefore we can substitute $p(\epsilon)$ for $q_\phi(\mathbf{z}|\mathbf{x})$, when considering their infinitesimals, so when the expected value is needed for optimisation, the independent marginal distribution p can be substituted into integral and we integrate over the known ε instead of the latent code, making the estimator differentiable. This substitution process can be applied to both networks, and thus the gradient optimisation algorithm of the trainer's choice has differentiable estimated posteriors to work with [31].

2.2.5 Diffusion Model

A diffusion model begins with real data and, in many steps, gradually adds noise and learns how the data changes. When the data turns to white noise, it will try to reverse the knowledge of how the data changes when adding noise to the original data. After many of these training steps, we can feed white noise to the model, and it will produce a new realistic output [51].

2.2.6 Normalizing Flow

A normalizing flow transforms a known probability density with multiple invertible mappings to resemble the frequency of occurrence in our training dataset. With this distribution, we can simulate new data points that statistically resemble the original dataset. [40]

2.2.7 Differential Privacy

To quantify the privacy loss, this report uses Differential Privacy[24] and Approximate Differential Privacy [17]. The privacy guarantee provided is that for some given data set D_1 and another data set D_2 which are different only by one data point(i.e. one person's contribution to the data set), and some synthetic data output or range $\mathcal{S}$ of a generating model $\mathcal{M}$

$$\Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq \exp(\epsilon) \times Pr[\mathcal{M}(D_2) \in \mathcal{S}]$$

This guarantee is achieved by adding noise to the generating model. Dvorak [24] constructs a privacy mechanism which samples this noise from a symmetric exponential distribution and adds it to the generating model. From this privacy mechanism Dvorak [24] also derives that ϵ is dependent on $\mathcal{M}$'s behaviour on D_1 and D_2 and on the variance σ of the noising distribution. One of the most useful properties of Approximate Differential Privacy [17] is the Post-Processing Theorem [16], since it ensures that no amount of generated data can compromise the (ϵ, δ)-differential privacy of the training data set. It states that for any function $g : \mathcal{S} \to \mathcal{S}'$ and a (ϵ, δ)-differentially private generative model M

$$\Pr[g \circ M(D_1) \in \mathcal{S}'] \leq \exp(\epsilon) \times Pr[g \circ M(D_2) \in \mathcal{S}'] + \delta$$

for the two data sets D_1, D_2 as defined above. δ denotes the probability that the privacy protection fails.

2.3 Image Data

Image data was the earliest data type synthesised in the more recent research we focused on [18], the introductory paper for GANs, show experimental results evaluated on the MNIST[2] and CIFAR10[1] data sets, and with a publication date of 2014, it predates a supermajority of other work discussed in this report. As such, most discussion of the challenges in generating image data now focuses on creating images of greater fidelity/photorealism and greater scale. Two works that discuss these challenges and focus on them specifically are Brock, Donahue and Simonyan(2019)[8] and Azizi et al.(2023)[5]. Both however generate in a non-private manner.

2.3.1 Generative Adversarial Network Based Methods

In general, we do not observe a uniform approach on where to apply gradient clipping and noising on the discriminator or generator. The post processing theorem guarantees that the gradients of only one of the components need to be changed, because the other component can then treated as the subsequent process.

2.3.1.1 DPD-InfoGAN and DP-InfoGAN

The authors propose two separate but related models. Building upon the work of Chen et al. (2016)[14], they introduce a differentially private training framework for InfoGAN, called DP-InfoGAN and another differentially private and distributed training framework DPD-InfoGAN. The rationale to work with InfoGAN lies in the fact that the feature output can be controlled by additional parameters which can be learned alongside the known training process for GANs. Differential Privacy is ensured in the training of a single InfoGAN client, by employing gradient clipping as proposed in [3] and noising gradients with Gaussian noise where the variance is a function of hyperparameters and the privacy budget, as expressed by the noise scale σ_n, on the discriminator's gradients. The discriminator's weights are updated with the Adam statistical optimisation function[30] applied to the gradients after noising and clipping. To enable the client to interact with the shared Q network, it is then used to estimate the posterior probability of some code condition on the data. These codes are the parameters with which to influence the feature space of the generated output. The associated loss function, here negative log likelihood is also calculated. Afterwards we backpropagate the updated weights to the generator and update the generator weights. We do not need to noise/clip the generator's gradients, because they fulfill DP by the post processing theorem.
In the distributed setting the clients update one shared auxiliary network's Q values and discriminator output in a round-robin fashion, while the next client in the sequence receives the already partially updated Q network values, to train their own model with them. This approach reduced the transmitted data per round and client, as the model parameters can stay local and need not be transmitted, saving on network operations necessary.

2.3.1.2 DP-BEGAN

The authors of this paper[43] build upon the work of Berthelot, Schumm and Metz[6] who proposed Boundary Equilibrium GAN. The chief difference between the GAN model proposed by Goodfellow et al. [18] is that the discriminator is specifically an auto-encoder neural network. The training framework DP-BEGAN[43] is not exclusive to the model and architecture but also encompasses operations on the training set. The sensitive training data is first clustered using the k-means++ algorithm, and from those clusters random samples are taken. The samples generated by this procedure are used to non-privately pre-train the model. During fine tuning, the actual data set is used.

2.3.2 DP-LFLow

Jiang and Sun(2023)[25] note that recent state of the art generative models(from 2019[45], 2018[27] and 2020[10] respectively) produce good output only at high levels of privacy loss $\varepsilon = 10$ and cannot scale on small ε. Smaller generative Models (tested on VAEs) have been shown to be more utility preserving under noise perturbation[52], which is an integral part of DP-SGD, and have lower training costs. This observation is reversed in non-private training setups. Latent Flow is used in this generative model to limit the model size enough to limit training expenditures, but maintain a standard of utility. The proposed model is composed of a normalising flow and an autoencoder. Unlike the basic flow model, the model proposed here is trained on the latent code of the autoencoder, with the training aim of simultaneously minimising the autoencoder loss and the normalising flow loss. An earlier work[41] had shown that image semantics are preserved even under heavy compression. This enables the autoencoder to be tuned ”aggressively”, reducing the latent space dimensions and thus also overall model size. Further, the training set is partitioned according to labels, train one instance of the proposed model on a subset of the training data. This subset is unimodal, since it was partitioned by label. This allows for the trained model to be smaller in size while maintaining the standard of accuracy of the larger model trained on the entire set. This partitioning is proposed because in the training process with labels (i.e. training a conditional model, which is state of the art at time of publication of the paper in question) DP-SGD also perturbs the labels, which improves no privacy-preserving properties, but makes training more cumbersome, and the authors thus choose to avoid this. The final model is the union of all the smaller models.

2.3.3 Autoencoder Based Methods

2.3.3.1 DP-AuGM

DP-AutoencoderGenerativeModel (DP-AuGM) is one of the two models proposed in Chen et al. [12]. The encoder network of the auto-encoder generative model[13] is trained on private data using DP-SGD. By its nature, it learns to encode distinguishing data features of the private data into a latent space. Simultaneously a decoder network is trained to reconstruct the higher-dimensional private data from the latent space. Using the l_2 distance between encoded sample and the latent space decoding, both neural networks are improved. Since the decoder could potentially leak private data, it is not used further than necessary to train the encoder network. The encoder is the part of this model that will be made publicly available and the authors also propose that it could possibly be integrated into Machine-Learning as a Service (MLaaS) or federated learning settings. A user of DP-AuGM would supply their own (potentially much smaller) data set as input to DP-AuGM's encoder, which then generates synthetic data samples from the user's own data. Those can then be used to train the user's own machine learning model for whatever downstream task they might be interested in. DP-AuGM is empirically shown to protect against three types of attacks against privacy preserving generative models: Membership Inference Attack, Model Inversion Attack and GAN-Based attacks.

2.3.3.2 DP-VaeGM

DP-variational auto-encoder based GenerativeModel (DP-VaeGM) is the other model proposed by Chen et al. [12]. Differences between the two proposed models lie in the differences between auto-encoders and variational auto-encoders, in that both the encoder and decoder both incorporate a latent variable z which is directly sampled from some external distribution $d(z)$, in this work taken to be gaussian: $\mathcal{N}(0,1)$. DP-VaeGM is experimentally shown to be effective against membership inference attacks. The model actually consists of n variational autoencoders, which are specialised to learn only one class of data, learned with a differentially private training algorithm. This multitude is shown in the paper to yield higher generation utility. The ultimate generation result is achieved by taking the union of all the generated samples from each model $M_i, i \in \{1, \dots, n\}$.

2.3.4 Diffusion Models

Their key contributions the authors Dockhorn et al. [15] identify pertain to the parameterisation of the Diffusion Model(DM), the sampling algorithm used to select the output, and noise multiplicity, a modification to the traditional DM training process, where "[a] single training data sample is reused for training at multiple perturbation levels"[1]. They identify their key contribution to be that the model size can be kept quite small while maintaining output quality, which provides many advantages for differentially private training settings, chiefly that many more training epochs can be afforded while not exhausting the privacy budget. This is hugely desirable for Diffusion Model Training. A lengthy section is devoted to the motivation to use Diffusion models: The authors firstly remark that the output quality of a diffusion model is on par or better than GAN output. Their loss function is like the L_2 error function in statistical regression models, and thus a lot simpler to estimate and understand. This also results in a more robust model that scales up and down more easily, and is crucially less vulnerable to the input perturbation of DP-SGD, and thus less likely to suffer model collapse. Another aspect that makes DMs superior to other generative models is the fact that the denoiser neural network takes multiple denoising steps to generate a data point, allowing the output to gradually approximate the statistical distribution instead of needing to accomplish the transformation in one generation step. This affords the denoiser to have a simpler architecture, smaller size and smoother functions. The stochastic sampling process also contributes to the improved quality of the model output, since the score model learned in the training process will be

[1] Direct quote from [15]

imperfect due to training with DP-SGD anyway, and stochastic sampling compensates. The objective function of a non-DP DM relies on many training iterations to counteract the noisiness of the distribution estimator. Since many training iterations mean a large privacy cost, DPDM instead averages over a number of estimators to control the noise inherent in the loss and added by the DP-SGD training procedure. The neural network is 2 orders of magnitude smaller than non-DP DMs. This keeps the loss down, since that scales linearly in the number of parameters. The authors of this paper prove in appendix B that DPGEN[11] does not guarantee differential privacy of its synthetic data.

Ghalebikesabi et al. take a different approach to Dockhorn et al.'s model minimalism. They instead propose to use standard diffusion models with an adjusted training methodology specifically for the differentially private context. The proposed and experimented with model has 45 times as many parameters as DPDM's model. Their methodology consists of the following points:

- Pre-training on a proven data set. It is later experimentally shown that even large variances in distribution between the pre-training set (which may even be public) and the fine tuning set do not have a noticeable impact on generation quality.
- augmentation multiplicity and timestep multiplicity. De et al. (2022) proposed using augmentation multiplicity, where each input sample of a batch is augmented (stretched, cropped, rotated).

2.4 Text

Deep recurrent models like those we discuss in this section have become very important for Natural Language processing and can be integrated in a wide variety of applications but most of them can involve sensitive information like passwords names or addresses, also a person could be identified implicitly by rare and unique phrases. Thus Natural language processing has to focus on the underlying patterns because attacks on models have shown how important privacy is [35].

2.4.1 Transformer Model

The transformer model, introduced by Vaswani et al [46], uses multi-headed attention to parallel-process the input text thus improving efficiency. There are multiple attention heads that operate independently. This helps the model to consider different aspects of the information from multiple lower dimensionality sub-spaces at the same time. This enables the transformer to understand complex relationships within the text by combining insights from multiple perspectives given by the attention heads. This architecture is crucial for improving the model's ability to deal with long-range dependencies and complex contexts [46].In chat GPT (Generative Pre-trained Transformer) the model in pre-trained on a extensive dataset thus making it much faster to process a request. Also it uses Few shot learning where in addition to the prompt examples are given on how the task can be solved [9]

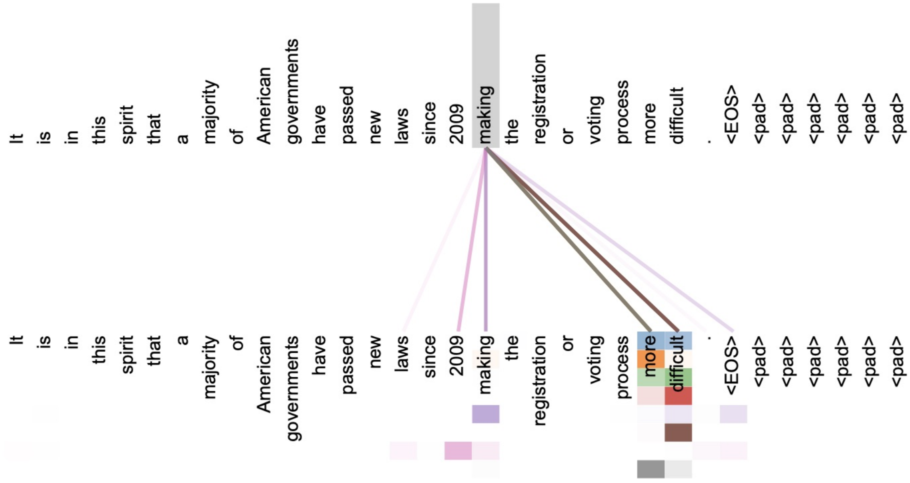


Figure 2.2: This visualization shows the distant relationships in the encoder's attention heads. Many of the attention heads, represented by colors, of the word making point to "more" and "difficult," thus having "making...more difficult" as a part of the finalized sentence.[46]

2.4.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs)[26] operate by processing sequences through loops within their architecture; with this, they keep information from previous steps in memory. Each neuron in an RNN considers input from data in the present step and the output from the neurons of the previous step. With this architecture, the model can consider the right next step, considering the context of the previous output, leading to a coherent output. The challenge, however, is when sequences get longer, RNNs become less efficient at capturing dependencies due to problems like vanishing or exploding gradients, meaning the weights either diminish or become too large to manage effectively as the model goes through each time step during training. This is where Long Short-Term Memory Networks (LSTMs)[22] try to solve this by implementing gates that regulate the flow of information. There are input, output, and forget gates; these allow the model to retain only crucial information and forget the rest, which improves the stability across long sequences. Therefore, the newer information retains its impact, and the older information might get forgotten depending on its importance. This makes them highly effective for natural language and efficiently manages the limited context window with what matters. [22] However, this provides no privacy guarantees; therefore, a PrivateRNN[35] was introduced that implements several strategies to protect user data from being compromised. The main principle used is Federated Averaging with Noise. This uses federated learning to update a global model with a user's own machine learning model and not the user's data. Next the users' models are averaged leading to abstraction of the data provided to the global server. Also, differential privacy tries to limit the privacy loss.It does this by gradient clipping and noise addition.

2.4.3 DP-RVAE

The Differentially Private Recurrent Variational Autoencoder (DP-RVAE) is a model proposed by Y. Wang et al [48] that aims to generate text while adhering to privacy guarantees defined by differential privacy. The model combines the traditional Variational Autoencoder (VAE) with a Recurrent Neural Network (RNN) that deals with the sequential nature that VAEs are inherently incapable of addressing. The DP-RVAE encoder transforms the text into a lower-dimensional latent space that contains the core features using an RNN. The decoder, which is also based on an RNN, then considers the latent space and the text that has already been generated to ensure it is coherent and is relevant for the posed query. Differential privacy is added as a two-step process: first, the model employs noise addition to obscure the data, and gradient clipping is used to limit the impact a single data point can have on the output of the model. The model output is then evaluated using BLEU and ROUGE that determine the quality of the text for coherence and fluency. DP can be used to measure the privacy loss that the model comes with.

2.4.4 Additional techniques

2.4.4.1 Prompt tuning

The Robust and Private Tuning (RAPT)[33] framework tries to modify prompts for privacy preservation, allowing large language models to be customized without compromising too much privacy. RAPT uses Privatized Token Reconstruction; this works alongside the main LLM training task. It involves creating a privatized token, which is the real data but with added noise or token shifting. This secondary task is then to reconstruct the original data from the private data, in this way the model learns the underlying patterns and does not focus on the privacy-compromising specifics. This also makes the model more resistant to variations in data representation.

2.4.4.2 Vickrey Mechanisms

Xu et al. [50] propose a novel approach to privacy-preserving text generation by adopting a mechanism that works similarly to the Vickrey auction system. This method introduces switches between choosing the most probable and the second most probable text output. Traditional differentially private (DP) mechanisms just add noise, and this can still leak data, especially if the noise is not enough. Xu et al.'s approach mitigates this risk by adding some additional probability into the generation and not always relying on the strictly better solution. This mechanism increases randomness, strengthens privacy and maintains the usability of generated text, but of course, the privacy has some cost. This mechanism is most valuable for applications in natural language processing that deal with private information.

2.5 Tabular Data

There are multiple challenges in synthetic tabular data laid out by Xu et al. in [49]:

- Real-world tabular data often contains discrete and continuous data columns. In order to generate either discrete or continuous output, the generator applies either $softmax$ for discrete values or $tanh$ to map the output $[-1, 1]$. So to generate both forms of data, both need to be applied to the data at some point.
- Continuous values in tabular data often do not follow a normal distribution. GANs however model continuous variables with Gaussians, so appropriate approximations need to be made. Further some might even follow multimodal distributions, which GANs are known to model poorly, with a tendency to leave out some modes entirely [44]
- High imbalance of values that can lead to the omission of minor classes in the GAN model because to the discriminator, it only makes a minor difference if they are included.
- Discrete data columns are encoded with one-hot encoding vectors, which may be very sparsely populated. Generators fill the corresponding synthetic data column using $softmax$. In early training epochs those synthetic columns may produce highly populated vectors, by nature of the generator training process. The discriminator network can then distinguish synthetic from real data by the sparseness attribute of the vector alone and not by comparing and the synthetic and real probability distribution. That in turn means that the discriminator does not need to learn the real probability distribution, which is a training failure.

2.5.1 Synthetic Data Vault

The Synthetic Data Vault [38], proposed by researchers from MIT, is an early approach to privacy-preserving synthetic tabular data. It outlines the main steps for creating synthetic tabular data. Each table is generated independently to simplify modeling and handle distinct distributions, allowing the model to focus on one distribution at a time. After generating the individual tables, conditional parameter aggregation is used to meaningfully link the independent values. It does this by identifying and conserving the relationship between the different fields. For categorical data, the model maps the continuous distribution to values based on the frequency of appearance of their respective categories. For time values, the model uses a distribution of seconds since the last epoch. To preserve privacy, noise is added to the data.

2.5.2 DP-Conditional GAN

The approach presented by in [49], is to modify an existing tabular data generator by embedding it in a privacy preserving training framework. For this, ConditionalTabular-GAN(CTGAN), proposed by Xu et al [49] was chosen. The authors explain that they chose to use CTGAN to develop their framework around, because it mitigates most of the challenges that generating tabular data pose to GAN models[49]. The framework employs the Moments privacy accountant[3] to enforce (ε, δ)-Differential Privacy, and the stochastic optimisation algorithm Adam[30] to improve the gradients of the generator and discriminator networks. The architecture of the generator and discriminator are taken from CTGAN, with the modification that the discriminator's gradients are clipped and Gaussian noise is added. The authors follow the reasoning of [54], and add that because only the discriminator has access to the sensitive data, noising only it's gradients is sufficient to fulfill the requirements of differential privacy. Further, the authors note, noising only the discriminator is advantageous for model stability, convergence and privacy loss estimation, for one, because noise is only introduced once, but also because the discriminator's architecture is the simpler one and is thus easier to estimate.

The authors also introduce a framework to train DP-CTGAN in a federated setting, which they name Federated DP-CTGAN(FDP-CTGAN). The framework allows each client participating in training to retain their data and train a copy of the DP-CTGAN model on that locally stored data, with the procedure lined out for a centrally trained DP-CTGAN. Every round the model is initalised with parameters given from the central server, and every round the parameters for generator and discriminator are sampled and aggregated, to obscure the contribution made by any individual client to preserve data privacy. Those aggregated parameters are averaged and approximate the true data distribution in the central server. The thusly updated parameters are then redistributed as the initial model for a new round of training.

2.5.3 DP-HFlow

Differentially Private Heterogeneous Flow [32], or DP-HFlow, is a model for generating synthetic tabular data. It uses four key technologies:

- Variational Dequantization: Normalizing Flow normally requires continuous data; therefore, we add noise to convert discrete values into continuous ones and thus allow for the Flow model to work.

- Conditional Spline Flow: This reduces the model complexity by using transformations multiple times across the model, which lowers the amount of parameters and simplifies the model.

- Fine-Grained Gradient Clipping: Per Unit Clipping: Adjusts clipping thresholds for individual units depending on their gradient, allowing important signals more influence.

- Stochastic Sparsification: Randomly sets smaller gradients as zero; this lets the model focus on significant features and reduces noise.

2.6 Time Series Data

Time series data consists of sequences of datapoints recorded at a certain time interval, and this is used in many industries. However, time series data also has its challenges, which stem mostly from its sequential nature, high dimensionality, and the temporal cohesion between datapoints that are often hard to discover. Additionally, it has datapoints that require more anonymization than others. [28]

2.6.1 Landmark Privacy

Landmark Privacy [28] improves differential privacy for time series by categorizing the data into landmark events and normal events. The landmark events represent very important events that are very sensitive in nature. For these landmark events, more strict privacy measures are used. This optimizes privacy protection while still not abstracting less important features away.

First, the landmarks need to be identified within the time series and marked as landmarks. This can be done using different metrics. Then the privacy budget is allocated; the landmarks receive a smaller portion of this budget to ensure they are properly protected, and the non-landmarks can benefit from a less strict abstraction, thus preserving their important features. The privacy budget is applied uniformly over the landmarks to ensure consistent privacy; also, dummy landmarks are inserted that aim to confuse potential attackers. One of the main challenges of this model is the difficulty of identifying useful landmarks, and the binary categorization of landmarks can lead to poor data quality or security for values that lie in between.

2.7 Location and Trajectory Data

Location traces or trajectory data are time series[36], and thus share many of the challenges present in generating time series. But location data exist in a three dimensional semantic space, the temporal component because it is a time series, a spatial component by the nature of the type of data under consideration, and a thematic component [39]. These three components' interrelation must be preserved through the entire process of synthetic data generation to provide any utility at all. Wang et al [47] capture the semantic space in four statistical metrics: Length distribution, diameter distribution, a measure of distance between individual Points of Interest (POI), trajectory density, the number of trajectories in a location/area, and the transition pattern, which Narita et al [36] capture in a transition matrix.

2.7.1 DP-TrajGAN

The approach described consist of 4 steps. The first step is necessary data pre-processing and creating a grid representation of trajectories across a map. In the second step the privacy budget is initialised. In the framework, the desired privacy budget is a predefined number ε. Since the semantics of a trajectory have both a spatial and a temporal component, both must comply with this bound. Therefore this model works with both a spatial privacy budget ε_s and a temporal privacy budget ε_t. The utility of the generated trajectory is directly correlated with the allocation of the available ε-privacy budget, and thus the model needs to find an optimisation of how to distribute the available budget between ε_s and ε_t. To that end the authors propose a sub-budget ε_s^i per map grid cell r_i, with ε_s^i being representative of the density of trajectories of all users per cell r_i relative to the trajectory densities in all other cells r_j. Temporal privacy is calculated on a per-user basis. The day is gridded into half-hour intervals. For each user u the privacy budget is the relative frequency of u being in a grid cell c for a time range $range$ accumulated over all time steps and grid cells. Thus, an initial allocation of the privacy budget is decided. In a second step, a partially observable markov decision process fine tunes the allocation of the privacy budget. States in this POMDP are characterised by a position on the privacy-utility spectrum and actions increase/decrease ε_t or ε_s once per step. The optimal location on that spectrum is evaluated via a reward function consisting of a measure of privacy, Mutual Information, and a measure of utility, chosen to be the Jensen Shannon Divergence. In every evaluation step the real data set is disturbed by a reallocated ε_s and ε_t, and the process terminates with either the optimised budget allocations or after a pre-determined number of iterations. These privacy budget parameters are then used to disturb the real data set one last time, to get a training set where DP guarantees can be given by post processing theorem. The disturbed data set is then used to train a GAN which models trajectories. The finished generator is then capable of generating differentially private trajectory data [53]

2.7.2 Co-Location

The approach presented in [36] offers a refinement strategy for existing generative models. Leveraging the fact that human social connections (the paper considers friendships specifically) cause humans to spend some time of the day in the same location as another (co-location). Some subset of the points in the trajectory can thus be determined based on co-locations with friends, and can be complemented with synthetic trajectory data, generated by some suitable process. In this work, the authors create a mechanism to synthesize differentially private co-locations based on a differentially-private friendship relation. These co-locations are then unified with synthetic traces generated by the mechanism presented in [7], by use of the Viterbi algorithm. These steps do not compromise differential privacy by the post-processing theorem. Both differentially private co-location Q and friendship probability p are realised via user level privacy. In the context of the p's graph representation, this means that some random nodes and all their associated edges are hidden. For the Co-location matrix user-level privacy hides entire location traces. The noise mechanism applied to Q is Laplacian, while for p the authors leverage the tree structure that results from the categorisation and sub-categorisation of different points of interest (POI) which the trace goes through. Privelet is a Wavelet transform that guarantees differential privacy, and employs the tree structure of POI categorisation to apply its noise. The authors chose Privelet over simple Laplacian noise, since it gives the same

ε guarantees at much lower noise levels per time instant and POI, improving synthesis quality [36].

2.7.3 Markov Decision Process Based Model

Some of the authors collected for this work don't propose a new model architecture directly, but rather an algorithm that may improve some aspects of the training process and accuracy of the model where their proposal is implemented. One such work is PrivTrace[47].

2.7.3.1 PrivTrace

Building on previous work, AdaTrace [20] and DPT [21], the authors use markov chain models(MCM) to learn the transition distribution in the data set. Their contribution is constructing both a first and second order MCM, to allow the trajectory patterns to be extracted more accurately than AdaTrace, for which the second order MCM is used, but not introduce noise that degrades the utility of the synthetic data, as occurred with higher order models used by DPT. After the training process, the model selection is based on the variation of predicted future trajectories from a given state i, indicated by the number of transitions N_i. If there are few future states, then the first order MCM is selected, since noise impacts the second order MCM more, and with fewer trajectories the noise makes up a larger part of the generated trace. If the variation of trajectories is large enough in both models, then the ratio $\frac{N_{i,1}}{N_{i,2}}$ is compared to a threshold. If $N_{i,1}$ is a lot larger, it is said to dominate, and thus the corresponding MCM should be used, else the second order MCM is used.

2.8 Genomics Data

As genomics is a field of rapid development, the availability of data is quickly growing; however, the data is often not shared because it is highly compromising [37], and even if data sharing is permissible, real data is very expensive [29] Therefore, synthetic data would be very useful. Unfortunately, the generation of high-quality synthetic data is very challenging. For this, the machine learning model must understand which features of the genetic code are privacy-compromising, like skin color or height, and which parts are functionally relevant and cannot be changed. Also the data has high dimensionality, which further restricts the usefulness of data that are differentially private. Thus, it is very hard to design a system that is privacy-preserving while still producing a useful output. Furthermore due to the nature of the data, not much training data for a machine learning model is available, further restricting development [37]. Because of the limited availability of research, both models we explain try to create the data, and we have not yet found a model that produces privacy-conserving synthetic data for genomics. We suspect that this is due to the challenges of producing the data even without additional requirements.

2.8.1 FBGAN

The Feedback GAN extends the Wasserstein GAN architecture by adding a Function Analyzer that analyzes the generator's output according to biological functions and inputs the highest-scoring outputs into the discriminator as Real data and replaces the oldest values in the discriminator. This provides a guideline for the discriminator on what is acceptable and what is not and gradually all the real data in the discriminator is replaced by high scoring outputs from the generator.This helps in creating data that does not just look realistic but also fulfills desired properties and satisfies the laws of Biology and allows to guide the generator to create for example antimicrobial properties [19].

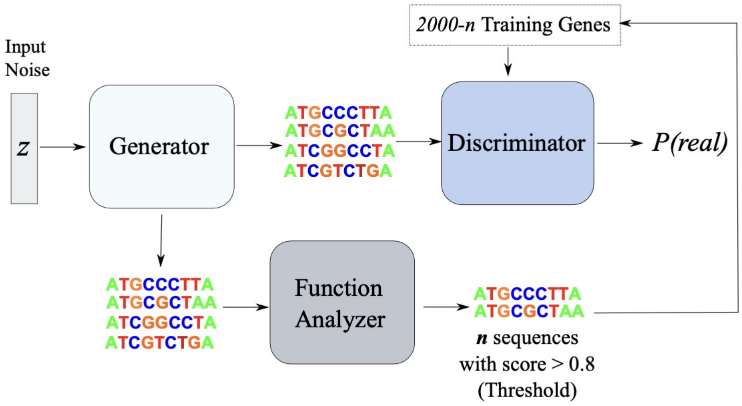


Figure 2.3: The feedback-loop training mechanism in FBGAN feeds the predictions sampled from the generator and inputs them into the analyzer. The analyzer scores each sequence and inputs the highest scoring sequences back into the discriminator as real data. n selected sequences from the analyzer replace the n oldest sequences in the "real" training dataset of the discriminator [19].

2.8.2 Deep Learning

A Deep Learning Neural Network refers to a architecture consisting of one input and output layer and Multiple Hidden Layers.This deep network allows it to even consider small nuances that a traditional model might consider noise and abstract away.Thus the network is much better at abstracting Non Linear data [34]. Deep Learning has shown to be able to learn the complex structure of DNA sequences and thus seems to be a promising technique for future Research. The biggest issues the explored deep generative model faced was a string motive match meaning a motive occurred more than once implying that privacy might be compromised easily [29].

2.9 Discussion

We believe that the future of synthetic data generation needs new, more refined models to make sure the training data stays private and secure. There is surely a need for models that go further in the direction of understanding the underlying patterns of the data without mimicking certain individuals. Also, the applied privacy preservation techniques are often rudimentary, relying on noise, gradient clipping, and introducing some decoys to confuse attackers. With sufficient effort, there could be potential to develop machine learning models from the ground up, integrating differential privacy into the core to optimize performance for specific data types, particularly in genomics. This field notably lacks models capable of generating high-quality, privacy-preserving synthetic data, thus highlighting a important research opportunity by combining insights from various models.

2.10 Conclusion

The exploration of privacy-preserving synthetic data generation in this report shows how synthetic data can be created used while minimising the risk for those in the learning database. Our taxonomy of synthetic data generation methods explores the landscape of methods, models, and frameworks that are used to create synthetic data. We also try to explain how the models add to the privacy protection while keeping in mind the tradeoff on data utility.

Acknowledgments

We sincerely thank Weijie Liu, our tutor, for his invaluable guidance and support throughout the writing of this seminar report.

2.11 Appendix

Paper	Performance	Main Takeaway	Quote
[37]	Enhanced genomic data security by balancing privacy and utility effectively.	Researched methods for Scoring and balancing privacy while considering data utility.	No single approach to generate synthetic genomic data yields both high utility and strong privacy across the board.
[29]	Achieved high accuracy in DNA design using deep generative models, though computationally intense.	Showed the potential of deep learning to advance biological design with accurate DNA sequence generation.	We show that our generative model can capture underlying structure in a dataset and manifest this structure in its generated sequences.
[19]	Optimized protein function precisely using a feedback mechanism despite model complexity.	Introduced a feedback loop in generative models to enhance bioengineering precision and functionality.	We have successfully developed a GAN model, FBGAN, to produce novel protein-coding sequences for peptides under 50 amino acids in length, properties.
[24]	Integrated differential privacy with ML, balancing privacy with performance.	Provided strategies for incorporating differential privacy into machine learning models, improving data security and user privacy.	We explore the interplay between machine learning and differential privacy, namely privacy-preserving machine learning algorithms and learning-based data release mechanisms.
[34]	Achieved high prediction accuracy in genomic analysis using deep learning.	Showed the potential of deep learning for genomic data analysis, leading to more accurate predictions.	-
[38]	Developed a framework for generating synthetic data with strong privacy guarantees.	Introduced methods for creating diverse synthetic datasets while preserving privacy, contributing to safer data environments.	For 11 out of 15 comparisons (>70%), data scientists using synthetic data performed the same or better than those using the original dataset.

Paper	Performance	Main Takeaway	Quote
[32]	Achieved privacy-preserving data synthesis using normalizing flows.	Showed the application of normalizing flows in generating private synthetic data, improving data privacy techniques.	The fine-grained gradient clipping methods proposed to train parameterheavy NF models are simple but shown to be highly effective in accelerating private training.
[25]	Developed methods for scalable and privacy-preserving image generation.	Integrated differential privacy into latent flow models, improving the privacy of generated images.	propose an effective solution, i.e. DP-LFlow, by reducing the flow training from the full input space to a lower dimensional latent space, so that the model is more resilient to (larger) noise perturbation introduced by DP-SGD
[13]	Achieved data privacy through autoencoder-based generative models.	Proposed a novel method using autoencoders to ensure differential privacy in synthetic data generation.	As we are the first to propose differentially private data generative models that can defend against the contemporary privacy violation attacks
[48]	Developed privacy-preserving text generation using VAEs.	Analyzed recurrent VAE architectures to generate high-quality text while preserving privacy.	-
[15]	Integrated differential privacy into diffusion models for high-quality data synthesis.	Advanced the field by adapting diffusion models to uphold privacy standards, enhancing the security of generated data.	-
[33]	Improved privacy in user interactions with large language models using prompt tuning.	Developed innovative prompt tuning strategies to enhance privacy, ensuring safer user interactions with large language models.	we propose Privacy-Preserving Prompt Tuning (RAPT), a framework for customizing and utilizing LLM service with privacy preservation. For privacy protection, RAPT applies a local privacy setting

Paper	Performance	Main Takeaway	Quote
[50]	Proposed models that balance privacy and utility in text generation.	Developed a utilitarian approach to improve both privacy and utility in generating text data, creating a balanced solution to privacy concerns.	A novel class of Vickrey mechanism is proposed, which not only enjoys metricDP but also optimizes the privacy-utility tradeoff within the constraint.
[23]	Enhanced privacy in time-series forecasting of health data using federated learning.	Analyzed methods for secure and private forecasting of sensitive health data making use of federated learning techniques.	-
[35]	Achieved competitive results with differentially private RNNs.	Showed the practical feasibility and effectiveness of training RNNs under differential privacy constraints to secure language model applications.	In this work, we introduced an algorithm for user-level differentially private training of large neural networks, in particular a complex sequence model for next-word prediction.
[28]	Developed configurable privacy settings for time series data, enabling effective protection.	Explored the significance and flexibility of customizable privacy settings developed specifically for the needs of time series data.	We presented landmark privacy for privacy-preserving time series publishing, which allows for the protection of important events while improving the utility of the final result compared to user-level differential privacy.
[53]	Developed GANs for realistic and privacy-preserving trajectory data generation.	Illustrated the integration of differential privacy into GANs to effectively secure trajectory data, while keeping it realistic and private.	DP-TrajGAN is used to generate synthetic trajectories close to the distribution of original trajectories, which can retain the statistical characteristics while protecting the user's spatio-temporal trajectory privacy.

Paper	Performance	Main Takeaway	Quote
[36]	Generated privacy-preserving location data with realistic co-location patterns.	Demonstrate the capability of synthetic data models to maintain both privacy and realistic spatial relationships in generated location data.	In this paper, we propose a novel location synthesizer that generates synthetic traces including co-locations between friends.
[47]	Applied adaptive Markov models to create private and high-utility trajectory data.	Explored how adaptive Markov models effectively balance privacy and utility, enhancing the quality and security of synthesized trajectory data.	In this paper, we propose a differentially private algorithm to generate trajectory data.
[3]	Implemented differential privacy in deep learning training processes.	Established foundational techniques for incorporating differential privacy into deep learning, securing the training process and protecting model data.	-
[12]	Developed autoencoder-based models for privacy-preserving data generation.	Reinforced the role of autoencoders in synthetic data generation with strong privacy protections, offering a good solutions for secure data handling.	-
[39]	Combined LSTM and GANs for effective privacy protection in trajectory data.	Demonstrate the interaction between LSTM and GAN technologies in enhancing privacy protections for trajectory data synthesis.	-

Bibliography

[1] ”CIFAR-10 and CIFAR-100. *datasets.* May 29, 2024. URL: https://www.cs.toronto.edu/~kriz/cifar.html.

[2] ”MNIST/IMAGES/GROUPS. *at master · mbornet-hl/MNIST”, GitHub.* May 29, 2024. URL: https://github.com/mbornet-hl/MNIST/tree/master/IMAGES/GROUPS.

[3] M. Abadi et al. “Deep Learning with Differential Privacy”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* Oct. 2016, pp. 308–318. DOI: 10.1145/2976749.2978318.

[4] J. Adejumo. “Gradient Descent From Scratch-Batch Gradient Descent, Stochastic Gradient Descent, and Mini-Batch dots”. In: *Medium* 12 (Apr. 2023). May 7, 2024. URL: https://medium.com/@jaleeladejumo/gradient-descent-from-scratch-batch-gradient-descent-stochastic-gradient-descent-and-mini-batch-def681187473.

[5] S. Azizi et al. “Synthetic Data from Diffusion Models Improves ImageNet Classification”. Apr. 2023. arXiv: Apr. URL: http://arxiv.org/abs/2304.08466.

[6] D. Berthelot, T. Schumm, and L. Metz. *BEGAN: Boundary Equilibrium Generative Adversarial Networks.* May 29, 2024. 2017. DOI: 10.48550/ARXIV.1703.10717. URL: https://arxiv.org/abs/1703.10717.

[7] V. Bindschaedler, R. Shokri, and C. A. Gunter. *Plausible Deniability for Privacy-Preserving Data Synthesis.* May 21, 2024. 2017. DOI: 10.48550/ARXIV.1708.07975. URL: https://arxiv.org/abs/1708.07975.

[8] A. Brock, J. Donahue, and K. Simonyan. “Large Scale GAN Training for High Fidelity Natural Image Synthesis”. 2019. DOI: 10.48550/arXiv.1809.11096. arXiv: Feb. URL: http://arxiv.org/abs/1809.11096.

[9] T. B. Brown et al. “Language Models are Few-Shot Learners”. 2020. arXiv: Jul. URL: http://arxiv.org/abs/2005.14165.

[10] D. Chen, T. Orekondy, and M. Fritz. *GS-WGAN: A Gradient-Sanitized Approach for Learning Differentially Private Generators.* May 29, 2024. 2020. DOI: 10.48550/ARXIV.2006.08265. URL: https://arxiv.org/abs/2006.08265.

[11] J.-W. Chen et al. “DPGEN: Differentially Private Generative Energy-Guided Network for Natural Image Synthesis”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* June 2022, pp. 8377–8386. DOI: 10.1109/CVPR52688.2022.00820.

[12] Q. Chen et al. “Differentially Private Data Generative Models”. 2018. DOI: 10.48550/arXiv.1812.02274. URL: http://arxiv.org/abs/1812.02274.

[13] Q. Chen et al. “Differentially Private Data Generative Models”. 2018. arXiv: Dec. URL: http://arxiv.org/abs/1812.02274.

[14] X. Chen et al. *InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets.* May 30, 2024. 2016. DOI: 10.48550/ARXIV.1606.03657. URL: https://arxiv.org/abs/1606.03657.

[15] T. Dockhorn et al. "Differentially Private Diffusion Models". 2023. DOI: 10.48550/arXiv.2210.09929. arXiv: Dec. URL: http://arxiv.org/abs/2210.09929.

[16] C. Dwork and A. Roth. "The Algorithmic Foundations of Differential Privacy". In: *FNT in Theoretical Computer Science* 9.3-4 (2013), pp. 211–407. DOI: 10.1561/0400000042.

[17] C. Dwork et al. "Our Data, Ourselves: Privacy Via Distributed Noise Generation". In: *Advances in Cryptology - EUROCRYPT 2006.* Ed. by S. Vaudenay. Berlin, Heidelberg: Springer, 2006, pp. 486–503. DOI: 10.1007/11761679_29.

[18] I. J. Goodfellow et al. "Generative Adversarial Networks". 2014. DOI: 10.48550/arXiv.1406.2661. arXiv: Jun. URL: http://arxiv.org/abs/1406.2661.

[19] A. Gupta and J. Zou. "Feedback GAN (FBGAN) for DNA: a Novel Feedback-Loop Architecture for Optimizing Protein Functions". 2018. DOI: 10.48550/arXiv.1804.01694. arXiv: Apr. URL: https://arxiv.org/abs/1804.01694.

[20] M. E. Gursoy et al. "Utility-Aware Synthesis of Differentially Private and Attack-Resilient Location Traces'". In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security.* Toronto Canada: ACM, Oct. 2018, pp. 196–211. DOI: 10.1145/3243734.3243741.

[21] X. He et al. "DPT: differentially private trajectory synthesis using hierarchical reference systems". In: *Proc. VLDB Endow.* July 2015, pp. 1154–1165. DOI: 10.14778/2809974.2809978.

[22] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://direct.mit.edu/neco/article/9/8/1735-1780/6109.

[23] S. Imtiaz et al. "Privacy Preserving Time-Series Forecasting of User Health Data Streams". In: *2020 IEEE International Conference on Big Data.* Atlanta, GA: USA: IEEE, Dec. 2020, pp. 3428–3437. DOI: 10.1109/BigData50022.2020.9378186.

[24] Z. Ji, Z. C. Lipton, and C. Elkan. "Differential Privacy and Machine Learning: a Survey and Review". 2014. arXiv: Dec. URL: http://arxiv.org/abs/1412.7584.

[25] D. Jiang and S. Sun. *DP-LFlow: Differentially Private Latent Flow for Scalable Sensitive Image Generation Transactions on Machine Learning Research.* May 21, 2024. June 2023. URL: https://openreview.net/forum?id=GEcneTl9Mk.

[26] M. I. Jordan. "Serial order: A parallel distributed processing approach". In: *Advances in Psychology.* Elsevier, 1997, pp. 471–495. URL: http://faculty.otterbein.edu/dstucki/COMP4230/Jordan-TR-8604-OCRed.pdf.

[27] J. Jordon, J. Yoon, and M. van der Schaar. *PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees.* May 29, 2024. Sept. 2018. URL: https://openreview.net/forum?id=S1zk9iRqF7.

[28] M. Katsomallos, K. Tzompanaki, and D. Kotzinos. "Landmark Privacy: Configurable Differential Privacy Protection for Time Series". In: *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy.* Apr. 2022, pp. 179–190. DOI: 10.1145/3508398.3511501.

[29] N. Killoran et al. *Generating and designing DNA with deep generative models.* May 21, 2024. 2017. DOI: 10.48550/ARXIV.1712.06148. URL: https://arxiv.org/abs/1712.06148.

[30] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". 2017. DOI: 10.48550/arXiv.1412.6980. arXiv: Jan. URL: http://arxiv.org/abs/1412.6980.

[31] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". 2022. arXiv: Dec. URL: http://arxiv.org/abs/1312.6114.

[32] J. Lee et al. "Differentially Private Normalizing Flows for Synthetic Tabular Data Generation". In: *Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 7*. June 2022, pp. 7345–7353. DOI: 10.1609/aaai.v36i7.20697.

[33] Y. Li, Z. Tan, and Y. Liu. *Privacy-Preserving Prompt Tuning for Large Language Model Services*. May 21, 2024. 2023. DOI: 10.48550/ARXIV.2305.06212. URL: https://arxiv.org/abs/2305.06212.

[34] J. Liu et al. "Application of deep learning in genomics". In: *Sci. China Life Sci.* 63.12 (Dec. 2020), pp. 1860–1878. DOI: 10.1007/s11427-020-1804-5. URL: https://doi.org/10.1007/s11427-020-1804-5.

[35] H. B. McMahan et al. *Learning Differentially Private Recurrent Language Models*. May 21, 2024. 2017. DOI: 10.48550/ARXIV.1710.06963. URL: https://arxiv.org/abs/1710.06963.

[36] J. Narita et al. "Synthesizing Privacy-Preserving Location Traces Including Co-locations". In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Ed. by J. Garcia-Alfaro et al. Cham: Springer International Publishing, 2022, pp. 20–36. DOI: 10.1007/978-3-030-93944-1\textunderscore2. URL: https://link.springer.com/10.1007/978-3-030-93944-1_2.

[37] B. Oprisanu, G. Ganev, and E. De Cristofaro. *On Utility and Privacy in Synthetic Genomic Data*. May 21, 2024. 2021. DOI: 10.48550/ARXIV.2102.03314. URL: https://arxiv.org/abs/2102.03314.

[38] N. Patki, R. Wedge, and K. Veeramachaneni. "The Synthetic Data Vault". In: *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Oct. 2016, pp. 399–410. DOI: 10.1109/DSAA.2016.49.

[39] J. Rao et al. *LSTM-TrajGAN: A Deep Learning Approach to Trajectory Privacy Protection*. May 24, 2024. 2020. DOI: 10.48550/ARXIV.2006.10521. URL: https://arxiv.org/abs/2006.10521.

[40] D. J. Rezende and S. Mohamed. "Variational Inference with Normalizing Flows". 2016. DOI: 10.48550/arXiv.1505.05770. arXiv: Jun. URL: http://arxiv.org/abs/1505.05770.

[41] R. Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models, 2021*. May 24, 2024. DOI: 10.48550/ARXIV.2112.10752. URL: https://arxiv.org/abs/2112.10752.

[42] S. Ruder. "An overview of gradient descent optimization algorithms". 2017. arXiv: Jun. URL: http://arxiv.org/abs/1609.04747.

[43] E.-M. Shi et al. "DP-BEGAN: A Generative Model of Differential Privacy Algorithm". In: *2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*. Shijiazhuang, China: IEEE, July 2022, pp. 168–172. DOI: 10.1109/ICCEAI55464.2022.00043.

[44] A. Srivastava et al. "VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning". 2017. arXiv: Nov. URL: http://arxiv.org/abs/1705.07761.

[45] R. Torkzadehmahani, P. Kairouz, and B. Paten. "DP-CGAN: Differentially Private Synthetic Data and Label Generation". In: *2019 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition Workshops (CVPRW). Long Beach, CA, USA: IEEE, June 2019, pp. 98–104. DOI: 10.1109/CVPRW.2019.00018.

[46] A. Vaswani et al. "Attention Is All You Need". 2023. DOI: 10.48550/arXiv.1706.03762. arXiv: Aug. URL: http://arxiv.org/abs/1706.03762.

[47] H. Wang et al. *PrivTrace: Differentially Private Trajectory Synthesis by Adaptive Markov Model.* May 21, 2024. 2022. DOI: 10.48550/ARXIV.2210.00581. URL: https://arxiv.org/abs/2210.00581.

[48] Y. Wang, X. Meng, and X. Liu. *Differentially Private Recurrent Variational Autoencoder For Text Privacy Preservation Mobile Netw Appl.* May 21, 2024. June 2023. DOI: 10.1007/s11036-023-02096-9. URL: https://link.springer.com/10.1007/s11036-023-02096-9.

[49] L. Xu et al. "Modeling Tabular data using Conditional GAN". 2019. arXiv: Oct. URL: http://arxiv.org/abs/1907.00503.

[50] Z. Xu et al. *On a Utilitarian Approach to Privacy Preserving Text Generation.* May 21, 2024. 2021. DOI: 10.48550/ARXIV.2104.11838. URL: https://arxiv.org/abs/2104.11838.

[51] L. Yang et al. "Diffusion Models: A Comprehensive Survey of Methods and Applications". 2024. arXiv: Feb. URL: http://arxiv.org/abs/2209.00796.

[52] D. Yu et al. *Do Not Let Privacy Overbill Utility: Gradient Embedding Perturbation for Private Learning.* May 29, 2024. 2021. DOI: 10.48550/ARXIV.2102.12677. URL: https://arxiv.org/abs/2102.12677.

[53] J. Zhang et al. "DP-TrajGAN: A privacy-aware trajectory generation model with differential privacy". In: *Future Generation Computer Systems* 142 (May 2023), pp. 25–40. DOI: 10.1016/j.future.2022.12.027. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167739X22004319.

[54] X. Zhang, S. Ji, and T. Wang. "Differentially Private Releasing via Deep Generative Model (Technical Report)". 2018. arXiv: Mar. URL: http://arxiv.org/abs/1801.01594.

Chapter 3

Federated Learning of Large Language Models (LLM) — A Review

Nicolas Huber, Nordin Dari

In recent years, significant breakthroughs in language models have made a revolutionary transformation possible by enabling the creation of large language models (LLMs). LLMs can now process and generate text with clear communication and adapt to various tasks. This breakthrough led to LLMs being used in various areas. However, their development is met with challenges due to the limited availability of public domain data and the need to maintain privacy concerning private domain data. This is where Federated Learning (FL) comes into play. Federated Learning allows entities to train collaborative models without sharing training data. This is particularly useful when multiple devices have a similar task but cannot share their local data due to privacy concerns. This led to the recent proposal of federated learning of large language models. In order to make FL of LLMs feasible on clients' devices, parameter-efficient fine-tuning (PEFT) algorithms have been developed. FederatedScope-LLM (FS-LMM) and Federated Instruction Tuning (FedIT) are two recent federated fine-tuning frameworks which implement PEFT algorithms. While PEFT algorithms reduce computational and communication costs in a federated setting, LLMs trained in a federated setting still lack performance compared to traditionally centralized-trained LLMs. Further challenges, among others, include 1) Heterogeneity in data quality, amount of data and computational resources on the client side. 2) The possibility of hostile actors injecting malicious training data into the federated learning process. 3) The possibility of reconstructing sensitive information through the broadcasted global model during federated learning, and 4) despite PEFT algorithms, the communication and computational requirements to perform FL of LMMs are, for most clients, still unacceptable.

Contents

3.1 Introduction

With the increasing capabilities of large language models (LLMs) in recent years, and especially the emergence of chatbots such as ChatGPT[1], it is only natural that both in the academic and industrial world, the interest in investigating and harnessing the potential of LLMs have been rising (Kuang et al., 2023). However, training LLMs needs huge amount of data and the urgently needed knowledge in certain domains, such as education, law or medicine, may not always be possible to share due to, for example, privacy concerns or country regulations [1] [14]. Federated learning (FL), which emerged in 2016 [22] seems to be a potential solution to this problem by enabling multiple clients to participate in the training process of a global model while preserving the privacy of the clients. In this report, we are first going to give a short introduction to centralized and decentralized FL, as well as some of their advantages and drawbacks. Afterwards, we will look at LLMs and try to foster an understanding of their workings and the advances over the years. Specifically, we try to understand the concepts of pre-training, fine-tuning, prompt engineering and the transformer architecture. Next, we will explore the combination of FL in LLMs, explore some of the algorithms that can be used to make the training process more efficient on the client side and understand some of the challenges that emerge when applying FL in LMMs. Finally, we take a look at two existing frameworks: Federated Instruction Tuning (FedIT) [32] and FederatedScope-LLM (FS-LLM) [14] which implement federated fine-tuning and federated instruction tuning. We will explore how the named frameworks work and what they consist of, as well as understand the evaluation of those frameworks and the trade-offs and challenges they found. In the end, we finish the report with an evaluation and discussion of the approaches discussed in this report.

3.2 Federated Learning

3.2.1 What is Federated Learning

Federated learning (FL) emerged in 2016 as a promising approach for multiple distributed clients to learn a shared model by aggregating locally-computed updates without the data leaving the client's device [22]. This is particularly useful since today more data points are stored where they are generated so there is more data available that is widely distributed but this makes data collection in central entities, as traditionally done in Machine Learning (ML), often impossible due to, for example, data privacy concerns or country regulations [1].

3.2.2 Decentralization Schema

Three different approaches[1] regarding the decentralization level of the federation architecture have been emerging, which can be seen in Figure 3.1. Namely Decentralized

[1] https://openai.com/blog/chatgpt

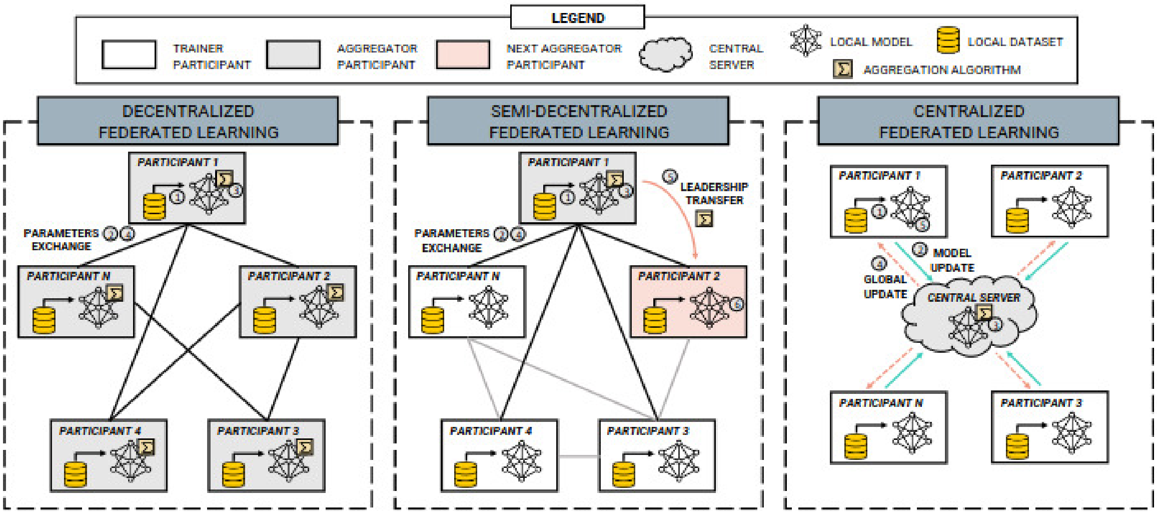


Figure 3.1: Three different decentralization levels of FL[1].

Federated Learning (DFL), Semi-Decentralized Federated Learning (SDFL) and Centralized Federated Learning (CFL)[1]. In DFL, the participating clients perform local model training, parameter exchange, local model aggregation, and parameter exchange again, independently[1]. In CFL, a central server is in charge of parameter aggregation and distribution, while the clients receive and update their local model accordingly and in SDFL, participating clients perform local model training and parameter exchange while an aggregator participant handles local model aggregation[1]. However, compared to CFL, in SDFL, the aggregator role rotates among the participating clients periodically by randomly selecting a neighbouring node of the current aggregator or choosing based on a client's network, computational or power capacity[1].

3.2.3 Centralized Federated Learning

In CFL, a central server creates and distributes a global model to all the participating clients. The clients train the model with their local data and, when they are done, send the local model parameters back to the central server, where the parameters get aggregated into the global model [1].

Since FL is a relatively new field, various challenges need to be solved for FL to be effectively applied in practice. Li et al. (2020)[16], mentioned the four core challenges, which include communication efficiency, systems heterogeneity, statistical heterogeneity and privacy.

Li et al. (2020)[16] point out that communication is a critical bottleneck in federated networks, given the potential scale of these networks and the need for every client to transmit individual data while maintaining privacy. However, they also propose a promising solution: the development of communication-efficient methods that send small model updates iteratively, rather than the model in its entirety.

Li et al. (2020)[16] further elaborate on the challenges posed by the different storage, computational, and communication capabilities of participating clients, as well as the potential unreliability of clients in a federated system. These complexities significantly complicate issues such as straggler mitigation, where some users take significantly more time to finish their assigned calculations, and fault tolerance [16].

Furthermore, Li et al. (2020)[16] explain that statistical heterogeneity becomes a challenge as clients generate and collect data in non-identically distributed manners, with significant variations in the number of data points across clients. This increases the likelihood of stragglers and adds complexity in terms of modelling, analysis, and evaluation [16].

Finally, privacy is another concern in CFL. To prevent adversaries from learning about the original training data based on analyzing the model parameters, algorithms need to be implemented to protect the privacy of clients. However, such algorithms come at the cost of increased computation [23].

3.2.4 Decentralized Federated Learning

In 2018 DFL emerged, where training data is distributed over many clients and the model parameters have to be shared and aggregated between the clients without a central coordinator [9].

Compared to CFL, DFL improves the robustness of the network since no central coordinator exists, therefore removing the need for the clients to put trust in the reliability of a central server for the training process [25]. Furthermore, DFL increases the flexibility and agility of a federated network, since regardless of clients joining or leaving the network, no reorganization is required, and the process can continue with minimal effort [25].

Despite the benefits of DFL, it also introduces new challenges, such as communication overhead and trust issues [1][31]. Ye et al. (2022)[31] point out that a reliable transportation layer protocol, such as transmission control protocol (TCP), is commonly used in a decentralized, federated network to offer reliable transmission between clients. However, this protocol may lead to communication overhead and reduce the possible number of connected clients [31]. Additionally, trust is an essential challenge which needs to be addressed. Careful consideration is required regarding which clients to let join the network, with which clients to share the model parameters and which clients model parameters one wants to aggregate into one's own model[1].

Furthermore, DFL allows for the relaxation of synchronous model updates since there is no central server in charge of parameter aggregation[1]. Depending on the topology of the network, issues such as fixed or dynamic topology arise[1]. The three topologies used in DFL are fully connected networks and partially connected networks, which include star-structured networks, ring-structured and random networks, and node clustering (see 3.2). Fully connected networks have the advantages of high reliability and robustness but do not scale well, and communication cost is high since if a new client wants to join the federation, it has to be connected to all the existing clients[1]. In star-structured networks, only one link to the proxy client, in charge of communication between the

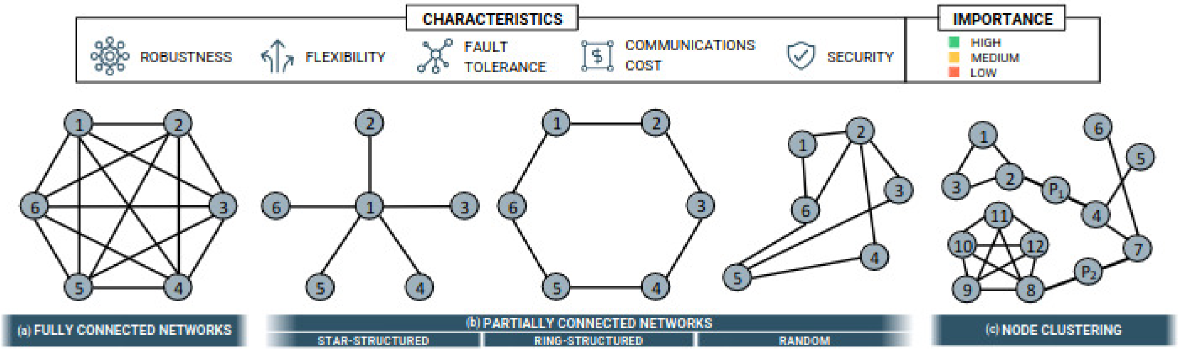


Figure 3.2: Different topologies of DFL[1].

clients, is needed to connect a new client to the network[1]. However, the central proxy client reduces flexibility, is a single point of failure and becomes a potential bottleneck since it has all the communication of the federation passes through it[1]. In a ring-strucuted network, communication costs grow linearly since each client maintains only two links, resulting in medium flexibility[1]. However, with a growing number of clients, transmission delay for model parameters increases[1]. An example of node clustering is similarity-based clustering, where clusters are determined by the similarity of the local model parameters of the clients, resulting in more individualized clusters for the clients that compose it, which leads to a homogeneous performance between nearby clients.[1]. Due to the clients within a cluster sharing similar data distribution, individually trained cluster models may be less generic and robust compared to models exposed to global data distribution[2][1].

3.3 Large Language Models

3.3.1 What is Language Modeling

As Liddy (2006, para. 48, [19]) defined:

> Language modeling is a statistical method for ranking documents in a collection based on the probability that they might have generated the query [...].

To understand how one can transition from unstructured data, such as text, to statistical inference we define some terms.

Token: A word or a subword. For generative applications this can also be symbols. For example the following sentence can be split up into tokens in such a way:

”This is not a lengthy sentence.” turns into:
[”this”, ”is”, ”not”, ”a”, ”length–”, ”–y”, ”sentence”, ”.”] .

We can see that ”lengthy” was split into ”length–” and ”–y”. This signifies that length can have multiple suffixes. So the meaning for an extent of something from end to end is encoded in ”length–” and the semantic context that can be added to that information is encoded in ”–y”.

Document: A coherent collection of sentences. This can be a book, a tweet, an article. Anything that logically belongs together. It is often assumed that a document is to be assigned a meaning or that a query (meaning) should generate relevant documents that statistically have the highest similarity to this meaning.

Corpus: A collection of documents. This can be a dataset of customer service conversations of a company where one conversation is one document.

Within this corpus one word can have a different meaning compared to other corpora. Delivery for Amazon means the transfer of goods. In a hospital context it can mean the application of medicine or birth of a child.

Embedding: A vector space representation of words. The representation of words is always in relation to the context. As per the previous example, delivery will be represented quite differently if the embedding is generated on just Amazons corpus or a combined corpus that also contains health care information.

3.3.2 Natural Language Modeling then and now

In today’s age with Chat-GPT and other generative models that create text that is hard to distinguish from human generated text it is hard to forget what language modeling used to mean how it relates to today’s state of the art models. Since human generated text is classified as unstructured data, classical statistical and ML approaches were not directly applicable to it. Before one could apply clustering or classification tasks on it, there needs to be preprocessing [7].

To transform words into a feature vector, first the text needs to be cleaned. Stemming is one techinique. The goal here is to reduce words to a base form. For example ”changing”, ”changed” and ”change” can get transformed to ”chang”. This conserves the meaning of the word while removing syntactic noise. Lemmatization does function similarly but does not reduce the words to a common stem but to a base form. The former examples would be transformed to ”change”. There is stop words removal where certain filler words like ”the”, ”a”, etc. get removed. One can see that those transformations were prescribed by a human. Such features in text used to involve quite a lot of human intervention before the wider adoption of self-supervised LLMs [7].

The resulting features were enabling frequency based algorithms like bag of words or term frequency - inverse document frequency.

One of the next steps was to embed sentences into a vector space with word2vec [27]. The embedding of words, sentences or documents into a vector space is also a concept that will be further utilized by large language models.

As one can see there are a multitude of preprocessing methods to translate the unstructured data into a form that allows for statistical analysis. This preprocessing also makes it a lot more difficult to generate legible text through inference since one removes lexical information to enhance the statistical significance. For over half a decade [11] the statistical approach was applied to machine translation which had some success but was qualitatively not comparable to translations by human experts. In the early 2000s there was ongoing work to apply shallow feed forward networks to this problem. From 2013 on Recurrent Neural Networks (RNNs) were introduced [28] and with it the field of Neural Machine Translation (NMT). In 2015 model architecture was expanded to not only translate word for word with some context but to generate whole sentences [28]. Figure 3.3 depicts the growing influence of NMT.

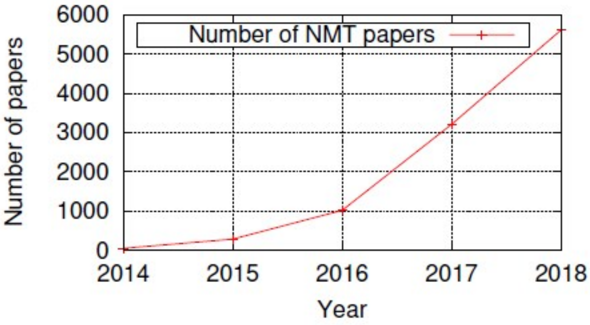


Figure 3.3: 2015 marked an inflection point for papers mentioning ”Neural Machine Translation” according to Google Scholar. [28]

Modern neural-network-based methods were based on model architectures such as Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM). These methods, compared to traditional machine translation methods, provide a stronger modeling ability and achieved better performance. One major downside is still the fixed size of of some parameters. This means longer sentences were difficult to be accurately translated[12].

3.3.3 Transformers

The 2017 break through paper ”Attention is all you need” [29] introduced the transformer architecture. This network architecture is an auto regressive encoder-decoder network that applies a novel method coined attention. The training for this network is self supervised. This means that many of the concepts for preprocessing, described in section 3.3.2, don’t need to be applied. This also removes some form of human bias from the training process.

3.3.4 Transformers Architecture

The transformer, as showcased in figure 3.4, is comprised of multiple elements. Firstly, with learned embeddings a token vector gets transformed into a representation in the

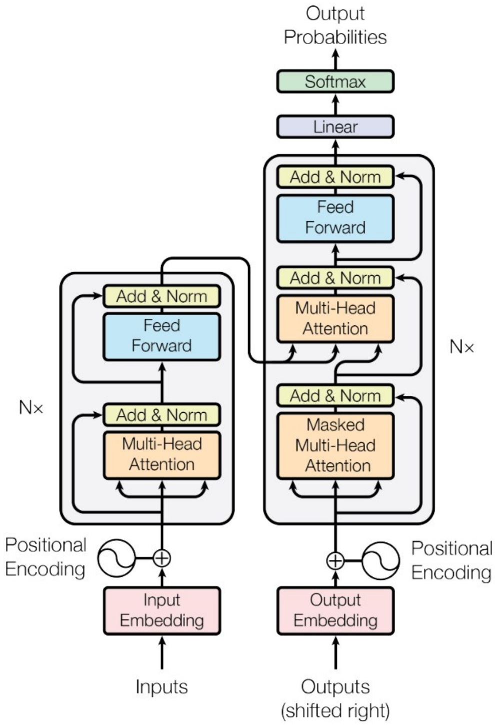


Figure 3.4: The transformer architecture as proposed by [29]. The left block showcases the encoder and the right block showcases the decoder.

vector space. Secondly, since this model does not convolve nor apply recurrence as other models described in section 3.3.2 it embeds the positional relation of words to each other.

Within the transformer blocks, of which multiple can be stacked, one finds attention blocks. These attention blocks return the relationship between encoded tokens and how relevant they are to each other as a continuous representation.

At the top of the decoder block output probabilities for the next word are generated. These are used to predict the next word in sequence which is then shifted and fed into the network again thus giving this architecture its augoregressive nature.

3.3.5 Training a Transformer

As defined in the section 3.3.3, transformers are training in a self-supervised manner. Multiple model classes and training strategies have emerged to set the weights within a transformer model.

Notably there are multiple optimization objectives for training an LLM [24][34]. (1) Masked language models (MLM) that need to reconstruct text that contains some masked tokens. (2) Autoregressive language models or next sentence prediction (NSP) models that need to correctly predict the next token or sentence in sequence. (3) Encoder-decoder or denoising autoencoders (DAE) models that first need to reconstruct masked/corrupted token vectors and then also predict then next token in a sequence. (4) Replaced token detection (RTD) that needs to determine if the current token has been replaced or not. (5) Sentence order prediction (SOP) that utilizes neighboring samples from a document to switch and determines the order.

Once defined how the model is trained the target is to use a large corpus to give the foundation model (Pretrained Foundation Model, PFM) a general idea about language. For some applications the corpus is specific for other the data sources are very general.

3.3.6 Fine-Tuning

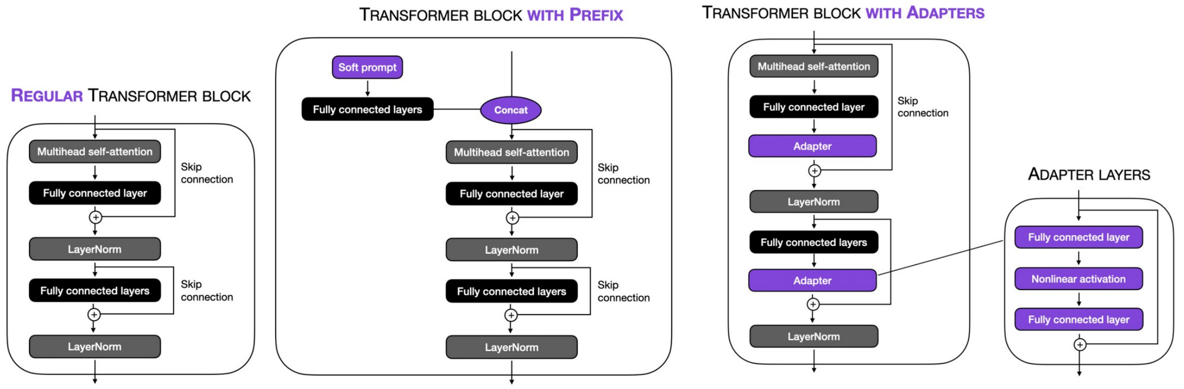


Figure 3.5: A PFM has its weights frozen and additonal, trainable layers are injected into the architecture. The additional layers are depicted in purple[26].

Fine-tuning a foundation model is akin to turning GPT-3 into ChatGPT. Several methods such as regular fine-tuning, adapter tuning, prefix tuning, Low-Rank Adaption (LoRA) [24]. The goal is to keep the generalized knowlege of language in the model but also improve its performance on one or several tasks. These methods generally work on allowing some or all parameters of the PFM to change.

Figure 3.5 showcases the difference in architecture for a regular transformer block to those with prefix-tuning and adapter tuning. Soft prompts are trainable layers that add an abstract vector to the input tokens while the network is instructed to train on a specific dataset. This ensures that the knowledge to infer this specific dataset well is encoded in the abstract vector. Adapters on the other side inject multiple trainable layers into the attention blocks[26].

Apart from modifying the network there are also ways to fine-tune a PFM by applying templates or context to the input text. Hard prompts can take the form of additional, human readable text, context that is added to the users prompts. Templates can also be used where the user is expected to ask a certain question that is then used to populate fields in a template so the PFM is better able to interpret the query [34].

With custom datasets one can apply different strategies for a model to be able to fulfill target tasks [13]. Few shot-learning, where a few examples of a given task are supplied to the model. One shot-learning where there is one example in the dataset that shall show the model how to act and zero shot-learning where there are tasks close enough to the target that allow the model to utilize knowledge from pre-training and context from fine-tuning to sufficiently be able to deliver answers when prompted for them. A couple of examples of prompts specifically crafted are shown in figure 3.10 and figure 3.11 of section 3.5.2.

3.4 Federated Learning in Large Language Models

Large Language Models have taken the world by storm. The datasets to train the ever growing models reaches multiple terabyte scale [21]. The availability of more data as well as the availability of data that is not available to the public can help improve language models abilities to generate better responses to prompts. There are two overarching reasons why this data is not available to the public or even just to entities that want to use it to pre-train or finetune their language models [3].

Privacy: Certain entities want to keep their data private because it contains information that is not publicly available and could give competitors and advantage or would cause the source entity to loose an advantage.

Regulation: Certain types of information can not be shared with anyone outside of the source entity. This can be either due to privacy guarantees like in medical data or can also be due to restrictions that certain data can not leave a country.

In spite of both of those reasons it might be interesting for a group of entities that possess data that is similar in nature to train models that shall perform similar tasks. It is also possible to enable one global entity to train a model while keeping the data within separate countries.

Some challenges one faces when combining Federated Learning and Large Language Models is the sheer model size. A 2023 survey [33] found 28 publicly available and 28 closed source models with a parameter count over 10 billion parameters. Table 3.1 shows a selection of models that has parameters in the billions or even trillions.

Due tue the nature of of LLMs concerning their size and need for data, centralized federated learning, as discussed in section 3.2.3, is feasible whereas decentralized federated learning, as discussed in section 3.2.4, is only feasible if all actors are able to contribute significant compute. This would not work for smaller devices as is often the case with DFL.

3.4.1 Federated Learning for Foundation Models

As discussed in section 3.4, pre-training a foundation model of a large language model would involve a lot of communication overhead between participating actors. Effectively synchronising model updates for each step is extremely expensive.

Name	Release Date	Parameters in Billions	Source
CPM-2	Jun-2021	198	Public
BLOOM	Nov-2022	176	Public
BLOOMZ	Nov-2022	176	Public
OPT	May-2022	175	Public
OPT-IML	Dec-2022	175	Public
GLM	Oct-2022	130	Public
Galactica	Nov-2022	120	Public
FLM	Sep-2023	101	Public
GLaM	Dec-2021	1'200	Private
PanGu-Σ	Mar-2023	1'085	Private

Table 3.1: Recent public and private LLMs have a very large parameter count[33]. This poses challenges to the infrastructure.

As utilizing a pre-trained foundation model and fine tuning it is generally the way to go we will discuss this in section 3.4.2.

An approach like pre-training a foundation model would only be necessary if there was a huge, fundamental change in regulations that would prevent cross-country exchange of text data. This is unrealistic since a large part of the corpora is scraped from publicly available web sources, available books or similar source available sources [21].

3.4.2 Federated Learning for Fine-Tuning

As discussed in section 3.3.6 there are several methods available to fine-tune a foundation model. If we compare some of the tuneable parameters shown in figure 3.5 to the final architecture of figure 3.6 wen can recognise that the model updates shared between entities are significantly smaller in opposition to updating the whole model as proposed for applying federated learning for foundation models.

Additionally to adapter and soft-prompt tuning [3] many of the common LLM finetuneinge techniques are applicable. Some of those are already implemented in in existing approaches as discussed in section 3.5.

3.4.3 General Threats and Remedies Proposed

Depending on the relationship between actors there are different risks involved with collaborative training [3]. During training, one has to be aware that there is a risk of model poisoning. This can be prevented or detected utilizing robustness aggregation or anomaly detection amongst others. It is also possible that a certain actor would know about a use case or task that is specific to another actor and tries to inject malicious examples that degrade the performance for this task.

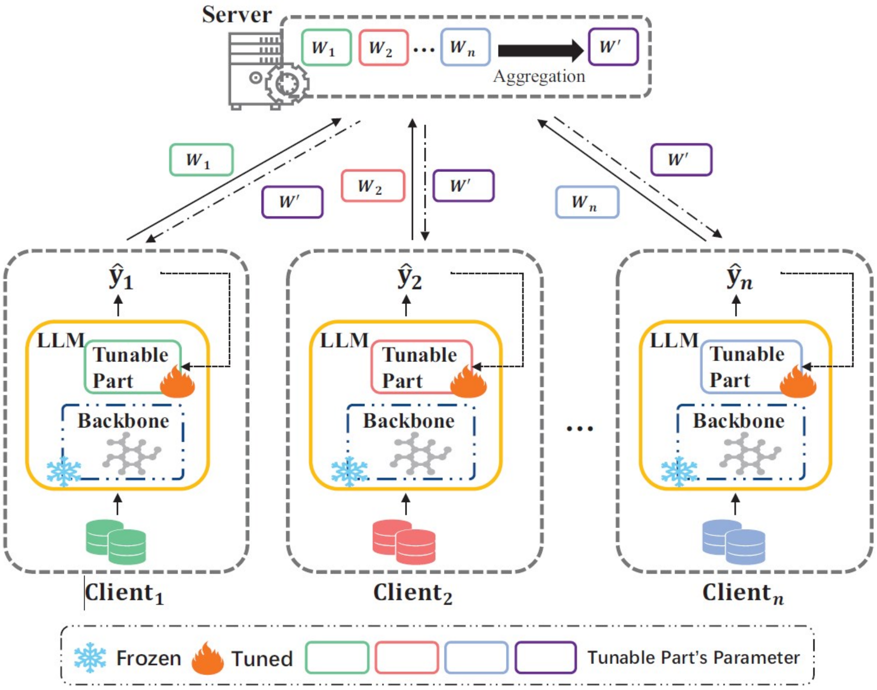


Figure 3.6: Fine-tuning a large language model with federated learning can be done utilizing a pre-trained foundation model where large parts of the network are frozen and only some of the original parameters or some additional parameters that are not part of the original architecture are trainable. [3]

It is possible to reconstruct data from just gradient updates [35]. Proposed remedies are adding noise to the gradients, increasing batch sizes or pruning gradients, meaning small gradient updates are set to zero which in turn should prevent the attacker to come to close to the real data.

3.5 Existing Approaches

Since FL, especially in combination with LLMs, is a rather novel research area there do not exist a lot of approaches to federated LLMs yet. Nonetheless, some frameworks have been developed. The goal of this section is to look at and understand two of the existing approaches, namely Federated Instruction Tuning (FedIT) and FederatedScope-LLM (FS-LLM) [14][32].

3.5.1 Federated Instruction Tuning (FedIT)

FedIT represents the first attempt to instruction tune LLMs in a CFL setting [32]. As demonstrated in figure 3.7, clients initially download a global LLM from a central server. Afterwards, the server performs client selection to determine which clients will participate in the current round of instruction tuning, then the selected clients perform instruction

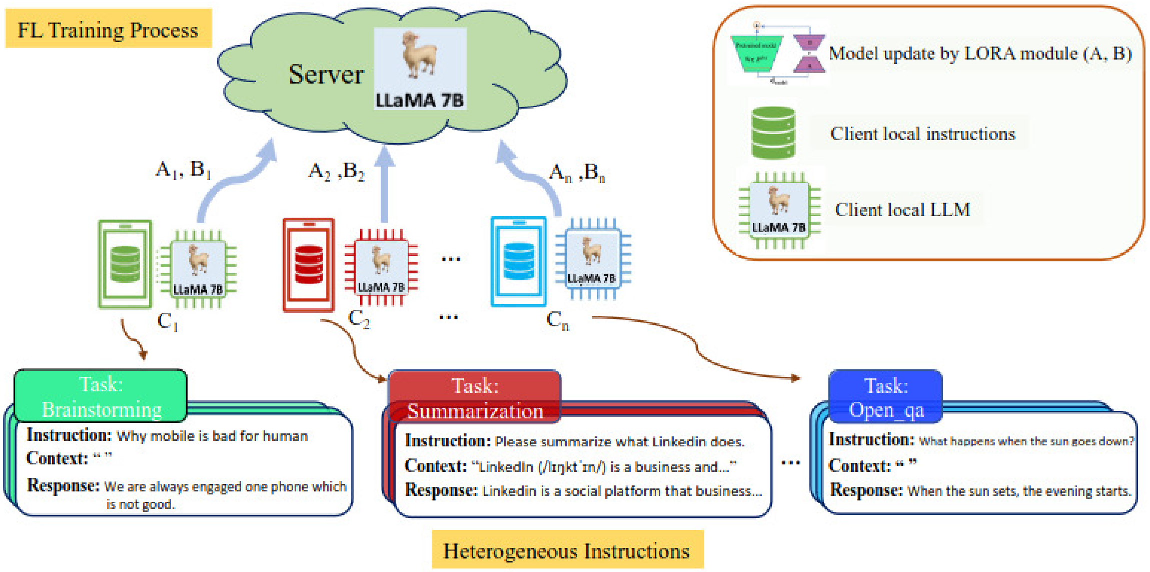

FL Training Process
Server
LLaMA 7B
Model update by LORA module (A, B)
Client local instructions
Client local LLM
A_1, B_1
A_2, B_2
A_n, B_n
C_1
C_2
C_n
Task: Brainstorming
Instruction: Why mobile is bad for human
Context: “ ”
Response: We are always engaged one phone which is not good.
Task: Summarization
Instruction: Please summarize what Linkedin does.
Context: “LinkedIn (/lɪŋktˈɪn/) is a business and...”
Response: Linkedin is a social platform that business...
Task: Open_qa
Instruction: What happens when the sun goes down?
Context: “ ”
Response: When the sun sets, the evening starts.
Heterogeneous Instructions

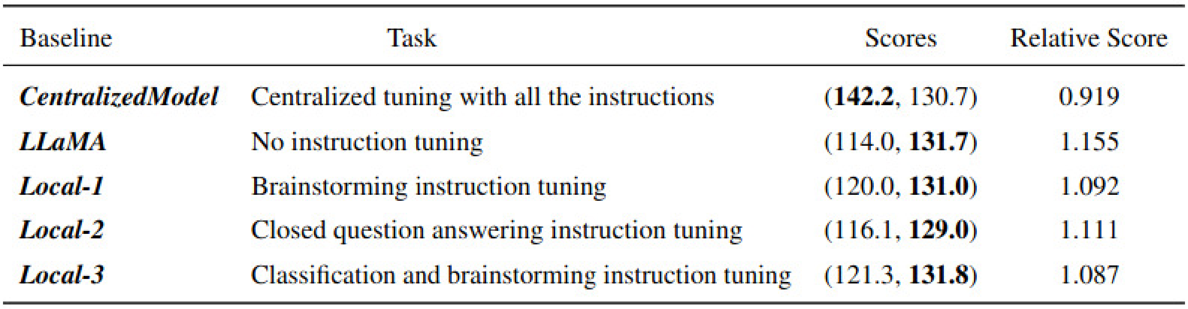
Figure 3.7: The Framework of Federated Instruction Tuning (FedIT)[32]

tuning on the global model with their local data and, when done, transmit the local updates back to the server, where they get aggregated back into the global model [32]. This process repeats a specifiable amount of times [32]. In order to adjust the learning process to the usually limited computational resources of the clients, FedIT employs parameter-efficient tuning methods, namely Low-Rank Adaptation (LoRA), which, according to Hu et al. 2021 [10], can reduce the number of trainable parameters by 10,000 times and the GPU memory requirements by three times [32].

According to Zhang et al., (2024)[32], FedIT illustrates a use case where statistical heterogeneity can be a positive factor for federated learning since the global model can be trained on a dataset of multiple clients where the content and format of the instructions can be substantially different from client to client. However, there is a need for further investigation of statistical heterogeneity in Federated Instruction Tuning regarding language diversity, particularly regarding fairness across underrepresented languages, domain-specific instructions, task complexity, task ambiguity, emotional tone, cultural factors and more [32].

To evaluate FedIT, Zhang et al. (2024)[32] implemented Shepherd, a framework to implement parameter-efficient federated instruction learning, and compared it to five baseline models. In their evaluation, Zhang et al. (2024)[32], assumed the presence of 100 clients, conducted 20 communication rounds, where each round 5 randomly selected clients performed one epoch of local training with their respective instruction dataset on a single Nvidia Titan RTX with 24GB memory.

As the global baseline model, Zhang et al. (2024)[32] used 7B LLaMA, an open-sourced LLM which has demonstrated performance on par with LLMs such as GPT-3. ”Local-1”, ”Local-2” and ”Local-3” have been fine-tuned on three different individual clients’ local dataset without model aggregation and ”CentralizedModel,” has been fine-tuned on all the



Baseline	Task	Scores	Relative Score
CentralizedModel	Centralized tuning with all the instructions	(**142.2**, 130.7)	0.919
LLaMA	No instruction tuning	(114.0, **131.7**)	1.155
Local-1	Brainstorming instruction tuning	(120.0, **131.0**)	1.092
Local-2	Closed question answering instruction tuning	(116.1, **129.0**)	1.111
Local-3	Classification and brainstorming instruction tuning	(121.3, **131.8**)	1.087

Figure 3.8: ”A summary of the baselines and their corresponding scores evaluated by GPT-4. The scores are reported in the format of (Baseline’s score, Shepherd-7B’s score) and the Relative Score is defined as (Shepherd-7B’s score / Baseline’s score)” (Zhang et al., 2024, p.10)[32]

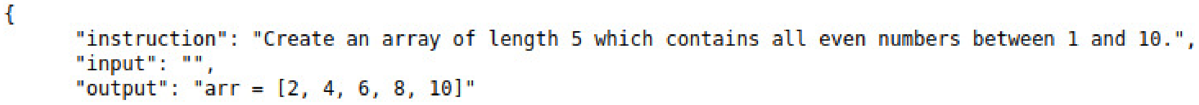
available data, representing the ideal training scenario where a central entity has access to the local data of the clients [32]. Zhang et al. (2024)[32] used GPT-4 to assess the generated responses of the different models, which the average scores of can be seen in figure 3.8.

The results demonstrate that FedIT effectively increases a model’s accuracy through federated instruction tuning, and statistical heterogeneity can be advantageous in a federated setting. The disparity to ”CentralizedModel” can, according to Zhang et al. (2024)[32], be partly attributed to the fact that the clients train the models locally with datasets that have substantially different distributions when compared to all the instructions as a whole. This might lead to the client’s local models learning different representations of the instructions.

3.5.2 FederatedScope-LLM (FS-LLM)

FS-LLM is an open-source package built on top of FederatedScope (FS), which aims to make federated fine-tuning of LLMs more efficient and less computationally expensive and provide a way to benchmark various federated fine-tuning LLMs algorithms [14][30]. Additionally, FS-LLM permits clients to fine-tune a closed-source LLM where the owner wants to keep the pre-trained LLM private from the clients [14].

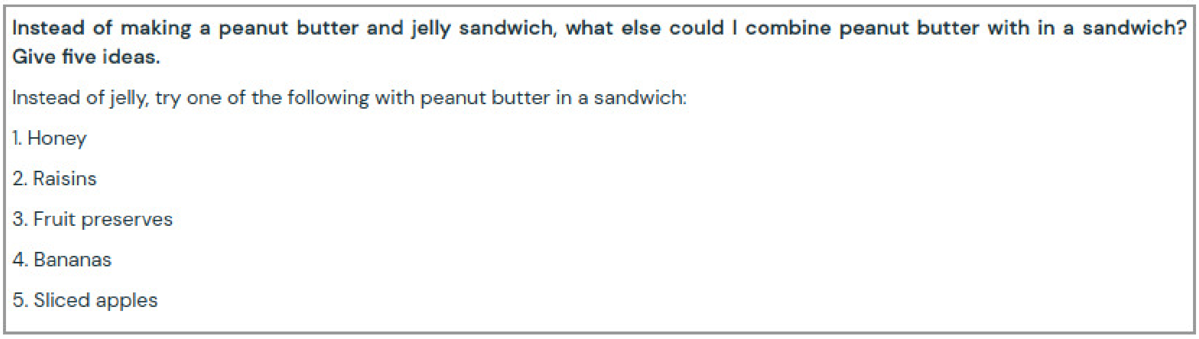
FS-LLM consists of the three modules: LLM-BENCHMARK, LLM-ALGOZOO, and LLM-TRAINER.



```
{
    "instruction": "Create an array of length 5 which contains all even numbers between 1 and 10.",
    "input": "",
    "output": "arr = [2, 4, 6, 8, 10]"
```

Figure 3.9: Coding Exercise from CodeAlpace[6]

LLM-BENCHMARK is the first module to evaluate LLM fine-tuning algorithms fairly and covers a complete benchmarking pipeline, including the construction of fine-tuning datasets and corresponding evaluation tasks [14]. It includes Fed-CodeAlpaca, Fed-Dolly and Fed-GSM8K-3. Fed-CodeAlpaca, built from CodeAlpace [6], stimulates a FL scenario with nine clients, where each client has one local fine-tuning dataset consisting of coding exercises with answers in one specific programming language [14]. figure 3.9 represents an example of such a coding exercise. The Fed-CodeAlpace dataset aims to enhance LLMs' code generation capabilities through fine-tuning [14].



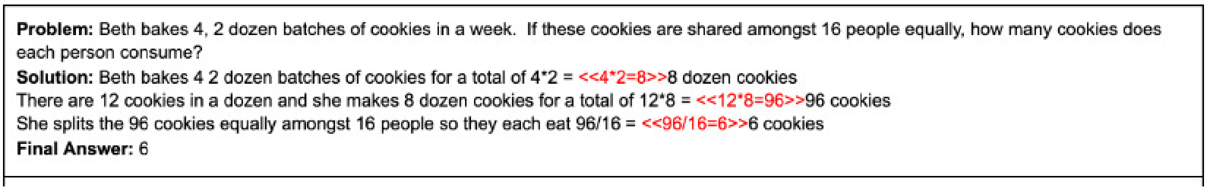
Instead of making a peanut butter and jelly sandwich, what else could I combine peanut butter with in a sandwich? Give five ideas.

Instead of jelly, try one of the following with peanut butter in a sandwich:

1. Honey
2. Raisins
3. Fruit preserves
4. Bananas
5. Sliced apples

Figure 3.10: Brainstorming Instruction from the Databricks-Dolly-15k Dataset [8]

Fed-Dolly represents the Databricks-dolly-15k dataset [8], partitioned into eight clients' local datasets, where each client's dataset only consists of one specific instruction category like, for example, the Brainstorming example seen in figure 3.10. Fed-Dolly aims to enhance the LLMs' capabilities for generic language through fine-tuning [14].



Problem: Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?
Solution: Beth bakes 4 2 dozen batches of cookies for a total of 4*2 = <<4*2=8>>8 dozen cookies
There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of 12*8 = <<12*8=96>>96 cookies
She splits the 96 cookies equally amongst 16 people so they each eat 96/16 = <<96/16=6>>6 cookies
Final Answer: 6

Figure 3.11: Math Question from the GSM8K dataset [5]

Finally, Fed-GSM8K-3 represents the GSM8K dataset [5], which consists of grade school math questions (e.g. figure 3.11), randomly partitioned into three subsets. Fed-GSM8K-3 tries to increase the capability of LLMs for chain of thoughts [14].

LLM-BENCHMARK also provides different splitters to partition the above-mentioned datasets into federated versions based on different meta-information or with different degrees of heterogeneity among clients [14]. Users can use these splitters to construct fine-tuning datasets, mirroring the usually inherent heterogeneity in FL settings [14].

In order to evaluate the potential improved capabilities of an LLM after a federated learning process, LLM-BENCHMARK provides three evaluation datasets, where each evaluation dataset can be used to evaluate the performance of federated fine-tuning through one of the provided fine-tuning datasets [14]. Specifically, HumanEval [4] is used to evaluate the improvements of LLMs in their capability of code generation, HELM [18] is used to evaluate the improvements of LLMs in their generic language capabilities and GSM8K-test

[5] is used to evaluate the improvements of LLMs in their chain of thoughts capabilities [14].

LLM-BENCHMARK also provides a set of cost-related metrics to measure the computation costs and communication costs of a federated fine-tuning process [14].

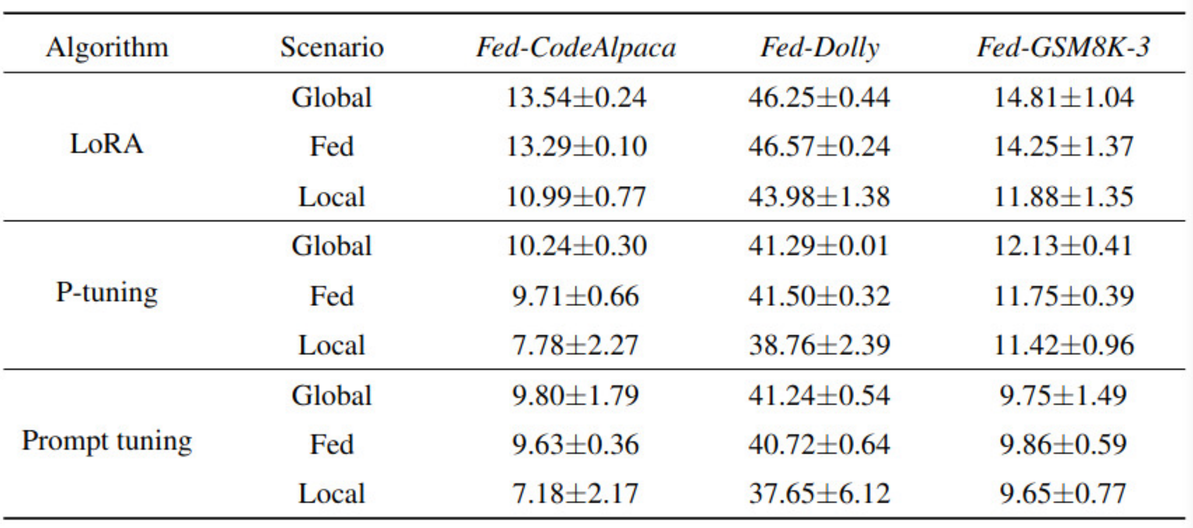
LLM-ALGZOO, the second out of three modules which make up the FS-LLM framework, provides a set of popular fine-tuning algorithms [14]. On the one hand, these fine-tuning algorithms consist of four parameter-efficient fine-tuning (PEFT) algorithms, which aim to reduce the communication and computation costs in federated fine-tuning in cases where all clients have access to the entire model. These four PEFT algorithms consist of LoRA [10], prefix-tuning [17], P-tuning[20] and prompt tuning [15]. On the other hand, for the cases where clients do not have access to the entire model but want to customize the model to their own liking, Kuang et al. (2023)[14] provide the algorithm FedOT. FedOT sends a lossy compressed model with untrainable parameters to the clients at the beginning of FL [14]. Clients are able to fine-tune adapters with their domain-specific data, while FedOT safeguards the intelligent property of the model providers as well as the data privacy of the clients [14].

To further optimize the federated fine-tuning process in terms of CPU/GPU memory consumption, multi-GPU parallel, and communication cost, Kuang et al. (2023)[14] provide LLM-TRAINER as the last building block of FS-LLM. LLM-TRAINER implements a set of accelerating and resource-efficient operators to reduce computation costs on the client side since, even with PEFT algorithms, fine-tuning LLMs in FL is computationally expensive for clients [14].

To evaluate the effectiveness of FS-LLM, Kuang et al. (2023)[14] conducted a set of experiments to evaluate how effective and efficient it is to federated fine-tune LLMs with PEFT algorithms and how effective it is to federated fine-tune LMMs without accessing the full model.

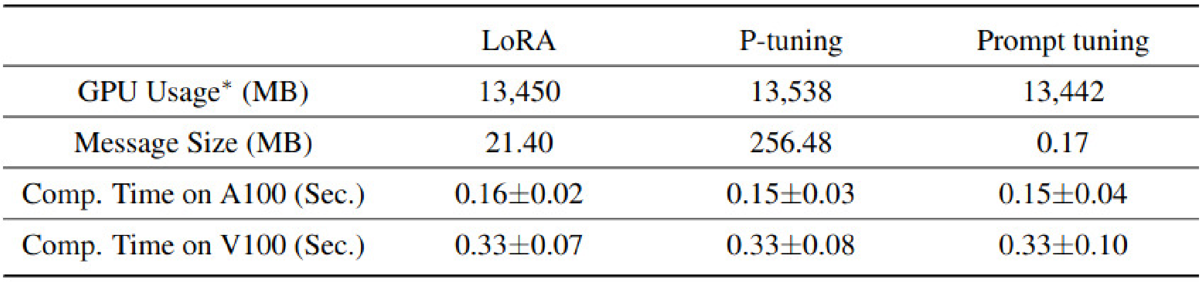
To benchmark the effectiveness and efficiency of different PEFT algorithms, Kuang et al. (2023)[14] conducted experiments with the three fine-tuning datasets, as well as the corresponding evaluation datasets provided by LLM-BENCHMARK on the LLaMA-7B model. All the experiments were conducted with a Nvidia A100 GPU (80GB) with Intel Xeon Platinum 8369B CPU and 512GB of RAM. Kuang et al. (2023)[14] repeated the experiments three times in all scenarios and reported the average evaluation score with its standard deviation, as can be seen in figure 3.12.

The global scenario represents one single client, who holds the entire fine-tuning dataset and performs fine-tuning on the LLM. The fed scenario represents federated fine-tuning where each client holds its own unique local fine-tuning dataset. And the local scenario represents a scenario where each client fine-tunes the LLM independently, with its own fine-tuning dataset. As shown in figure 3.12, all PEFT algorithms, when used in the federated setting, outperform the PEFT algorithms in the local scenario. So, it can be assumed that it is realistic and practical to use PEFT algorithms for fine-tuning LLMs in a federated setting [14]. Secondly, LoRA seems to be the most effective PEFT algorithm among the three listed.



Algorithm	Scenario	*Fed-CodeAlpaca*	*Fed-Dolly*	*Fed-GSM8K-3*
LoRA	Global	13.54±0.24	46.25±0.44	14.81±1.04
	Fed	13.29±0.10	46.57±0.24	14.25±1.37
	Local	10.99±0.77	43.98±1.38	11.88±1.35
P-tuning	Global	10.24±0.30	41.29±0.01	12.13±0.41
	Fed	9.71±0.66	41.50±0.32	11.75±0.39
	Local	7.78±2.27	38.76±2.39	11.42±0.96
Prompt tuning	Global	9.80±1.79	41.24±0.54	9.75±1.49
	Fed	9.63±0.36	40.72±0.64	9.86±0.59
	Local	7.18±2.17	37.65±6.12	9.65±0.77

Figure 3.12: Comparing the effectiveness of different PEFT algorithms when fine-tuning LLaMA-7B: Evaluation Scores(%) ± standard deviation(%)[14].

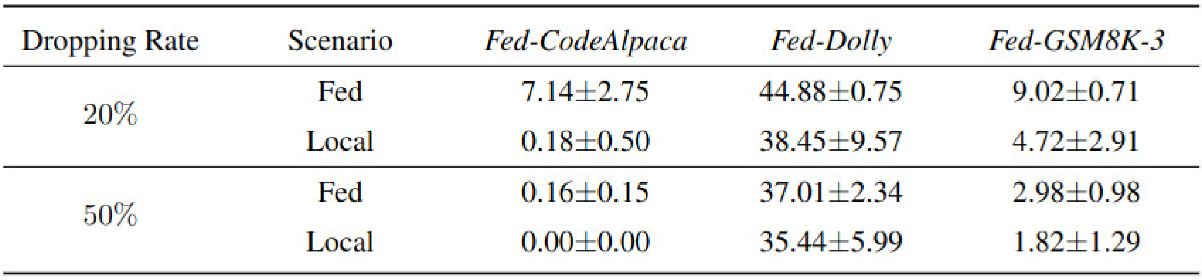


	LoRA	P-tuning	Prompt tuning
GPU Usage* (MB)	13,450	13,538	13,442
Message Size (MB)	21.40	256.48	0.17
Comp. Time on A100 (Sec.)	0.16±0.02	0.15±0.03	0.15±0.04
Comp. Time on V100 (Sec.)	0.33±0.07	0.33±0.08	0.33±0.10

Figure 3.13: Comparing the efficiency of different PEFT algorithms when fine-tuning LLaMA-7B in FL. Using Nvidia A100 GPU (80GB) and Nvidia V100 GPU (32GB): Evaluation Scores(%) ± standard deviation(%)[14].

When comparing the different PEFT algorithms regarding their efficiency, the results can be seen in figure 3.13 Kuang et al. (2023)[14] note that the significant difference in message sizes leads to large variations in transmission time. Furthermore, the computation time when the Nvidia V100 GPU (32GB) is used is roughly double the time then when the Nvidia A100 GPU (80GB) is used. From this, Kuang et al. (2023)[14] conclude that federated fine-tuning LLMs may suffer from more idle time due to the heterogeneity of computing resources among different clients. Therefore, Kuang et al. (2023)[14] highlight the need for two research directions emerging from this issue: (1) How is it possible to leverage the idle time of clients which possess superior computation-resources compared to other clients in the federated setting, and (2) how can clients with limited computation-resources efficiency utilizing available bandwidth during computation.

Finally, Kuang et al. (2023)[14] evaluate the performance of federated fine-tuning closed-source LLMs using FedOT. They used the first and last two layers of LLaMA-7B as the adapter and then compressed the model by dropping 20% and 50% of the remaining layers uniformly [14].



Dropping Rate	Scenario	*Fed-CodeAlpaca*	*Fed-Dolly*	*Fed-GSM8K-3*
20%	Fed	7.14±2.75	44.88±0.75	9.02±0.71
	Local	0.18±0.50	38.45±9.57	4.72±2.91
50%	Fed	0.16±0.15	37.01±2.34	2.98±0.98
	Local	0.00±0.00	35.44±5.99	1.82±1.29

Figure 3.14: Comparing the performance when fine-tuning compressed LLaMA-7B without accessing the full model under federated and local scenario: Evaluation Scores(%) ± standard deviation(%)[14].

Figure 3.14 demonstrates that through FedOT, multiple clients can benefit from federated learning when they cannot access the full model [14]. While FedOT achieves similar performance compared to some of the PEFT algorithms, Kuang et al. (2023)[14] note that this is due to the fact that FedOT sacrifices communication efficiency for model performance since the number of parameters of the adapter of FedOT is significantly larger than the number of parameters in the PEFT algorithms. However, when dropping 50% of the remaining layers, a chain of thoughts and code-generation capabilities acquire a minimal amount of new knowledge. So overall, FedOT appears to be a suitable choice when clients want to perform fine-tuning on closed-source models. However there is a trade-off between increasing the privacy of LLMs and degrading their performance [14].

3.6 Evaluation and Summary

The position paper [3] proposed pre-training, fine-tuning and prompt-tuning for the combination of federated learning and large language models. We propose that pre-training, in opposition to fine-tuning and prompt-tuning, is not feasible if one tries to utilize a competitively large model.

When evaluating the current approaches in section 3.5 we could not conclusively determine one method to be significantly better than others. As this field is still evolving a lot there will be improvements that have to be watched.

Due to the relatively recent nature of FL, especially in combination with LLM, studies and approaches regarding federated learning of large language models are still in their infancy. In this report, we have looked at two existing approaches to fine-tuning [32] and instruction tuning [14] LLMs in a federated setting. The laid-out approaches seem to be novel and heading in promising directions. Especially, the use of parameter-efficient fine-tuning (PEFT) algorithms seems to be a promising approach in order to reduce the computation resources needed for clients to participate in FL [14][32]. However, Kuang et al. (2023)[14] state that even with the use of PEFT algorithms, computation cost is still too high for most clients with limited resources. Additionally, FL for LLM suffers significant communication overhead with the amount of data transfer needed to rival centralized training performance [32]. In this sense, PEFT algorithms are a step in the right direction, but there is a need for computation-efficient algorithms that further tackle this issue.

FL enables clients to participate in model training without sharing their sensitive data, which is a significant breakthrough, particularly as unused public data becomes scarcer. Nonetheless, FL also introduces new challenges regarding privacy, which need to be addressed in further research. Specifically, preventing adversaries from joining an FL process with the intention of polluting the model by injecting crafter instructions [32], making it entirely impossible to reconstruct or recover sensitive client information through the broadcasted global model [32] as well as addressing the trade-off between model compression rate and model performance during federated fine-tuning of closed-source LLMs [14].

Finally, Zhang et al. (2024)[32] have shown that statistical heterogeneity can be advantageous in FL for LLMs. However, heterogeneity remains a complex challenge that requires attention, particularly when considering factors such as language diversity, domain-specific instructions, task complexity, emotional tone and cultural factors[32]. Moreover, the heterogeneity in computing resources among clients can lead to idle time for clients with computing-rich resources, raising the question of how to leverage such idle time in a federated setting [14].

3.7 Conclusion

The domain of combining federated learning and large language models is gaining traction, even tough there are still obstacles, like the explosive growth of model sizes, in the way. As the discussion after our seminar presentation has shown, interest in privacy preserving LLM training is definitely there. Also, different people have different ideas and boundaries that determine how much data they would provide for such a training and how they would like to be compensated. Unfortunately, in the current state people still would need to hand over their data to a trusted third-party that joins training efforts in their name as it is currently not feasibly for each user themselves to join the training efforts.

Bibliography

[1] E. T. M. Beltrán et al. “Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. IEEE Communications Surveys & Tutorials”. In: *vol.* 25.2 (2023), pp. 2983–3013.

[2] C. Briggs, Z. Fan, and P. Andras. “Federated learning with hierarchical clustering of local updates to improve training on non-IID data”. In: *2020 International Joint Conference on Neural Networks (IJCNN).* Ed. by P. 2020, pp. 1–9.

[3] C. Chen et al. “Federated large language model: A position paper”. arXiv preprint. 2023. arXiv: `2307.08925`.

[4] M. Chen et al. “Evaluating large language models trained on code”. arXiv preprint Juli. 2021. arXiv: `2107.03374`.

[5] K. Cobbe et al. “Training verifiers to solve math word problems”. arXiv preprint. Nov. 2021. arXiv: `2110.14168`.

[6] Code. *alpaca: An instruction-following llama model for code generation.* April 2024. URL: `https://github.com/sahil280114/codealpaca`.

[7] R. Collobert et al. “Natural language processing (almost) from scratch”. In: *Journal of machine learning research* 12 (2011), pp. 2493–2537.

[8] M. Conover et al. *Free dolly: Introducing the world’s first truly open instruction-tuned llm.* 2024. URL: `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`.

[9] L. He, A. Bian, and M. Jaggi. “COLA: Decentralized Linear Learning *Advances in Neural Information Processing Systems*”. In: *Vol.* 31, S.). Curran Associates, Inc (2018), pp. 4536–4546. URL: `https://proceedings.neurips.cc/paper_files/paper/2018/file/05a70454516ecd9194c293b0e415777f-Paper.pdf`.

[10] E. J. Hu et al. “Lora: Low-rank adaptation of large language models”. arXiv preprint June 2021. 2021. arXiv: `2106.09685`.

[11] J. Hutchins. “Machine translation: A concise history”. In: *Computer aided translation: Theory and practice* 13.29-70 (2007), p. 11.

[12] Y. Jia. “Attention mechanism in machine translation”. In: *Journal of physics: conference series* 1314.1 (Oct. 2019), p. 012186.

[13] S. Kim et al. “The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning”. arXiv preprint. 2023. arXiv: `2305.14045`.

[14] W. Kuang et al. “Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning”. arXiv preprint. Sept. 2023. arXiv: `2309.00363`.

[15] B. Lester, R. Al-Rfou, and N. Constant. “The power of scale for parameter-efficient prompt tuning”. arXiv preprint. Sept. 2021. arXiv: `2104.08691`.

[16] T. Li et al. "Federated learning: Challenges, methods, and future directions". In: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 50–60. DOI: 10.1109/MSP.2020.2975749.

[17] X. L. Li and P. Liang. "Prefix-tuning: Optimizing continuous prompts for generation". arXiv preprint Januar. 2021. arXiv: 2101.00190.

[18] P. Liang et al. "Holistic evaluation of language models". arXiv preprint. Nov. 2022. arXiv: 2211.09110.

[19] E. D. Liddy. "Document Retrieval, Automatic". In: *Encyclopedia of language et linguistics*. Ed. by K. Brown. Elsevier, 2005, pp. 748–755.

[20] X. Liu et al. "P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks". arXiv preprint. Oct. 2021. arXiv: 2110.07602.

[21] Y. Liu et al. "Datasets for Large Language Models: A Comprehensive Survey". arXiv preprint. 2024. arXiv: 2402.18041.

[22] B. McMahan et al. "& y Arcas". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by B. A. Communication-Efficient Learning of Deep Networks from Decentralized Data Singh B. A. Communication-Efficient Learning of Deep Networks from Decentralized Data Singh Aarti and Jerry Zhu. Vol. 54, Apr. 2017, pp. 1273–1282.

[23] H. B. McMahan et al. "Learning differentially private recurrent language models". arXiv preprint Oktober. 2017. arXiv: 1710.06963.

[24] B. Min et al. "Recent advances in natural language processing via large pre-trained language models: A survey". In: *ACM Computing Surveys* 56.2 (2023), pp. 1–40.

[25] Y. Qu et al. "Decentralized federated learning for UAV networks: Architecture, challenges, and opportunities". In: *IEEE Network* 35.6 (2021), pp. 156–162. DOI: 10.1109/MNET.001.2100253.

[26] S. Raschka. *Understanding Parameter-Efficient Finetuning of Large Language Models: From Prefix Tuning to LLaMA-Adapters*. lightning.ai - Creators of PyTorch Lightning, Apr. 2023. URL: https://lightning.ai/pages/community/article/understanding-llama-adapters/.

[27] X. Rong. "word2vec parameter learning explained". arXiv preprint. 2014. arXiv: 1411.2738.

[28] F. Stahlberg. "Neural Machine Translation: A Review and Survey". arXiv preprint. 2019. arXiv: 1912.02047.

[29] A. Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[30] Y. Xie et al. "Federatedscope: A flexible federated learning platform for heterogeneity". arXiv preprint. Nov. 2022. arXiv: 2204.05011.

[31] H. Ye, L. Liang, and G. Y. Li. "Decentralized federated learning with unreliable communications". In: *IEEE journal of selected topics in signal processing* 16.3 (2022), pp. 487–500. DOI: 10.1109/JSTSP.202.

[32] J. Zhang et al. "Towards building the federatedGPT: Federated instruction tuning. Zhang, Jianyi and Vahidian, Saeed and Kuo, Martin and Li, Chunyuan and Zhang, Ruiyi and Yu, Tong and Wang, Guoyin and Chen, Yiran (Eds.)" In: *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Apr. 2024, pp. 6915–6919. DOI: 10.1109/ICASSP48485.2024.10447454.

[33] W. X. Zhao et al. "A survey of large language models". arXiv preprint. 2023. arXiv: 2303.18223.

[34] C. Zhou et al. “A comprehensive survey on pretrained foundation models: A history from BERT to ChatGPT”. arXiv preprint. 2023. arXiv: `2302.09419`.

[35] L. Zhu, Z. Liu, and S. Han. “Deep leakage from gradients”. In: *Advances in neural information processing systems* 32 (2019).

Chapter 4

Inference Attacks on Machine Learning

Said Haji Abukar, Jonas Krumm

Contents

4.1 Introduction

Machine-learning is becoming an increasingly important tool in many aspects of our lives. While the rise of Large Language Models such as OpenAI's GPT models or Google's Gemini has brought machine-learning to public attention, it's domain extends beyond tasks of natural language processing. Machine-learning is being applied in a broad range of contexts; from Large Language Models such as ChatGPT and Gemini to guiding medical treatments. [2]

Machine learning profits greatly from the abundance of data available in today's digital landscape, popular training methods such as deep learning rely on vast amounts of training data to develop optimally accurate weight-distributions using methods of back-propagation and gradient descent [8]. Stemming from this prevalent use of vast data-sets arises, as with any use of data, an issue of confidentiality and privacy. Public discourse on machine learning is largely focused on resulting models and the utilities they provide. To the public eye it might then seem as though the data a model is trained on is inaccessible, seeing as only the models themselves are made directly publicly available. However, recent research shows that granting public access to a trained ML model might entail unintentionally publicizing the data it was trained on or other details about it's training environment [3, 16, 13]. With ML models being applied in contexts where training data is potentially very sensitive, such as in diagnostics tools in medicine, the ability to prevent leaking the training data while still allowing public access to the model is imperative. One of the primary threats in this regard is a class of attacks on ML models known as *inference attacks*. Inference attacks work by trying to extract some information about a model or its training environment by means of querying a model systematically, running its output against or comparing it with other models. Inference attacks are aimed at *inferring* information from model behavior.

In the following sections of this report we will cover different types of inference attack, categorizing these attacks by the kind of information they are aimed at extracting. We start with a short introduction of relevant information on machine learning and neural networks and some basic terminology for inference attacks. From there we move on to section 1.2 on inference attacks, beginning with a brief overview of different types of inference attacks before addressing them individually. Having introduced different kinds of inference attack, we then turn to countermeasures against these attacks.

This report is intended to serve as an overview, thus we will not delve into detailed implementations of the discussed attacks. Instead, we will provide broad descriptions focusing on key characteristics of these attack types.

4.1.1 Machine Learning & Neural Networks

In the present section, we provide a short preliminary introduction to the keywords neural networks, machine learning, and deep learning. We rely on a paper by LeCunetal [8] as the source for this section.

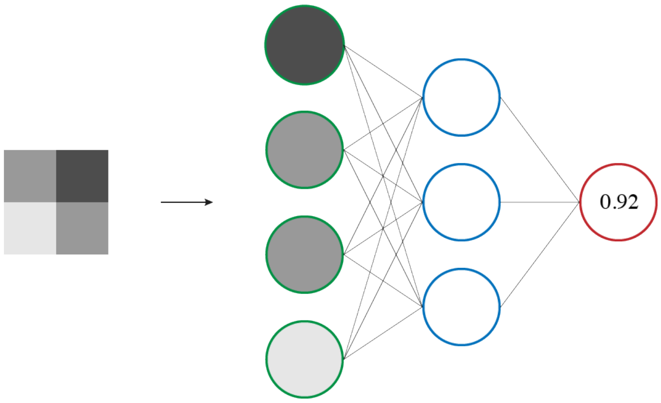


Figure 4.1: FCNN with input-, hidden- and output-layer

Firstly, let us touch on the concept of a neural network (NN). A NN is essentially a network of nodes or neurons as shown by figure 1.1. The figure shows a so called *fully connected neural network* (FCNN) where each node of every layer is connected to each node of the previous layer. There are other architectures where not all nodes are connected, but this is the fundamental concept. The NN in figure 1.1 includes three *layers*. The green nodes make up the input-layer, the blue nodes are a hidden-layer, and the red node is in the output-layer. Any layer between the input- and output-layers is called a hidden layer. NNs in real-world applications include several hidden layers and thousands of nodes in each of them.

The connections between nodes are weighted connections. To arrive at the value for a node n in the hidden-layer, all input nodes are multiplied by the weight of their respective connection to n and summed up. This repeats until the output-layer is reached, which might output e.g. a probability that the input image includes a rabbit.

Because this process of multiplication and addition would make the NN a linear function from input to output, so-called *activation functions* are used. Activation functions are simple non-linear functions that are applied to each node's value deciding whether the respective node is activated, i.e. taken into account in the calculation of the next layer. This allows NNs to model functions that are non-linear and thus more complex. A widely used example of an activation function is ReLU: $max(x, 0)$. To control the overall activation behavior of the NN, a *bias* can be added to nodes' values before the activation function is applied.

The different layers of a NN often correspond to different levels of representation: A first hidden-layer might detect edges present in an image, a second one might then detect shapes based on the inputs of the edges-layer etc. This process of abstracting raw data into a different representation is called *representation learning* [8, p. 436]. Deep learning, as defined by [8] is representation learning with multiple levels of representation, where the different levels of representation do not have to be manually engineered.

Deep learning uses an automated process of weight-adjusting to train NNs. Briefly summarized; a point of training data is ran through the network, from the NN's output an *objective function* is calculated, that measures the disparity between the actual and the expected output. To minimize this disparity, a gradient vector is calculated, that indicates how changing a specific weight would impact the objective function's output. The weights are then adjusted accordingly, and the process is repeated.

In contrast to how input is processed by the network, that is, in a *feed-forward* fashion, where the first layer passes data to the second and so on, deep learning uses a method called *backpropagation*. Weights are thus adjusted "from the back" i.e., from the output- to the input-layer. The gradient vector of a function at a certain point shows the direction of steepest ascent, i.e. how the parameters would need to be adjusted to increase the functions value the quickest. Gradient descent then, goes into the *opposite* direction of this vector, to minimize the functions output. By calculating the gradient of the objective function, we can determine what the output would need to look like to minimize it. The idea of backpropagation is that this can applied backwards. Having determined what the output layer would have needed to look like, one can then calculate what the output of the previous layer would have needed to look like to minimize the objective function and so on. Having calculated what all the layers' values would have needed to look like, it then becomes trivial to adjust the weights accordingly. One such adjustment is called an *epoch*, determining how many epochs a model should be trained for is a complicated question and outside the scope of this report.

Finally, an important aspect of machine learning with regards to inference attacks are *hyperparameters*. The term hyperparameters denotes all parameters that go into optimizing and training a model. This might include e.g., the *bias* applied to nodes before passing them to an activation function, the activation function itself, in which *step size* the weights are adjusted and so on.

4.1.2 Inference attacks - basics

Inference attacks are a specific type of attack directed specifically at *machine learning* (ML) models. Very generally, they aim at extracting some type of information from interacting with ML-models. The ML model that is under attack is often called the *target model*. As outlined, inference attacks are aimed at *inferring* some sort of information, this can broadly be split into two categories. Inference attacks can (a), be aimed at information about the *training data of a ML model*. Say a given model developed in the medical sector is trained on patient records and made publicly available via an API. An attacker might try to use inference attacks to extract data points from the training data i.e., in this case, to extract singular patient records indirectly by means of interacting with the ML model. The second category (b) is information about the architecture and hyperparameters of *the ML model itself*. As the industry of *machine learning as a service* (MLaaS) is continuously growing, ML models themselves are becoming more and more important business assets. Training a model capable of competing at market standard requires substantial investments into both gathering training datasets and fine-tuning model performance during training. The case where an attacker intends to steal the result of such investment, i.e. the resulting ML model, are thus becoming more critical.

A useful taxonomy to further categorize inference attacks was introduced by [10]. Using this taxonomy, even inference attacks of the same type can be sorted into different cases and analysed as such. The taxonomy considers inference attacks with regards to the type of access the attacker has to the target model itself, and the type of access the attacker has to the training data of the target model.

First, we turn our attention to the two types of access an attacker might have to the target model itself. An attacker either has *white-box* or *black-box* access to the model. White-box access implies that the attacker can not only query the target model, but also has access to it's architecture and the hyperparameters. Black-box access on the other hand, means that the attacker has access to neither of the two, i.e. they can only interact with the model through some predefined API and have no information about either the architecture or the parameters of the target model.

Now, let's turn to the second point of interest for the taxonomy introduced by [10], namely, the type of access the attacker has to the training data of the target model. Here, three distinctions are proposed by the paper. An attacker might thus have (1) *no dataset*, (2) a *shadow dataset*, or even (3) *a partial training dataset*. (1) means that the attacker has no access to the training data, (3) means they have a part of the actual training data available to them. A shadow dataset (2) is a dataset that mimics the *distribution* of the training data; an attacker would thus need to know the distribution of the data points the target model was trained on, to create such a shadow dataset. The important thing to understand here, is that the shadow dataset is usually made up of synthetic data. Thinking back to the example from a model in the medical sector: If an attacker had a shadow dataset of the training data, they would have a set of fake patient records that mimics the distribution of the actual patient records used to train the target model.

With the two points of interest mentioned, the taxonomy ends up with six combinations of white-box/black-box access and some type of dataset. The paper introducing the taxonomy does not regard cases where attackers have black-box access and no dataset, as it deems the success-rate of such attacks to be negligible [10, p. 2].

4.1.3 Relevance of the issue

In this section we try to emphasize the relevance of inference attacks as a possible threat to the fast growing field of MLaaS. We discuss the threats posed by inference attacks to machine learning, specifically that data-confidentiality is endangered which, if no effective countermeasures are found, could lead to far reaching restrictions on public access to machine learning models if they require confidential data to be trained.

4.1.3.1 Federated Learning

In federated learning a central model is trained across decentralized devices or servers. Instead of pooling all data at a central location, each participant in federated learning system can train a local model with their own data, and then share the resulting hyperparameters to the central model. The central model, which is accessible to all participants, is basically a combination of all the hyperparameters each participant arrived at. This has sevaral advantages; first and foremost, it allows the cooperative training of a ML model, without having to share the required data to all participants. For example, a single hospital might not have enough patient data to train their own model, but it can also not pool all patient data with 10 other hospitals for privacy reasons. In addition, federated learning overcomes infrastructure limitations that might be encountered by centralized machine learning. Federated learning is thus an important approach to enable machine learning in privacy sensitive domains (which are manifold).

Inference attacks pose a serious risk to the applicability of federated learning. In the case of membership inference attacks, exchanged gradients during FL training can be used to infer membership of data points in participant(s), e.g. when using a natural-language model a non-zero gradient of the embedding layer reveals the presence of words in training batches allowing adversaries to infer if certain text appeared in the training dataset of participants [11].

4.2 Inference Attacks

This section provides an overview of types of inference attacks, it categorizes them depending on whether they are aimed more at the data a model is trained with or at the model itself. Further, this section provides a description of each of the covered attacks and discusses the key considerations with respects to each type of attack as found in the relevant reviewed literature.

4.2.1 Attacks aimed at training Data

4.2.1.1 Membership Inference Attack [4]

Membership Inference Attacks (MIAs) focus on extracting sensitive information from the training data of the target models, e.g., age, personal preferences, health status. The inherent risks of such MIA are identity theft, reputation damage and loss of trust in ML-based systems.
MIAs exploit two primary forms of knowledge:

Knowledge of Training Data: Attackers exploit information about the distribution.
Often assuming access to a shadow data set that is known or data distributions. This knowledge enables the execution of non-trivial MIAs, assuming no overlap between the shadow and training data sets.

Knowledge of the Target Model: Understanding the training of the target model, including details of the learning algorithm, architecture, and learned parameters, will ensure attackers gain valuable insight into potential vulnerabilities.

4.2.1.2 Approaches to Membership Inference Attacks

MIAs employ various methodologies based on adversarial knowledge levels, with binary classifier-based and metric-based approaches being prominent.

Binary Classifier-based Attacks: These attacks aim to train a binary classifier to distinguish between a target model's behavior on its training members and non-members. They exploit overfitting in machine learning models, making them susceptible to inference attacks.

Training the Shadow Models: Multiple shadow models are created to mimic the target model's behavior. Leveraging the target model's structure and learning algorithm, shadow models are trained using disjoint shadow training datasets. This process captures the target model's behavior on similar data records.

Constructing the Attack Model: Prediction vectors obtained from shadow models are used to label data records as ”member” or ”non-member.” Subsequently, ”member” and ”non-member” datasets are constructed for training the attack model. This enables the attack model to distinguish between data records that were part of the training dataset and those that were not.

Metric-based Attacks: Metric-based Membership Inference Attacks (MIAs) employ various metrics derived from prediction vectors to infer the membership status of data records. These attacks mostly analyze the behavior of the target model by comparing metric values against predefined thresholds. There are four primary types of metric-based MIAs:

Prediction Correctness-based MIA inferring the membership based on whether the predicted membership status matches the actual membership status according to the base model.

Prediction Loss-based MIA inferring membership based on whether the prediction loss for a data record is lower than the average training member loss. Lower prediction loss suggests that the target model's prediction for the data record closely resembles its training behavior on member instances.

Prediction Confidence-based MIA inferring membership using the maximum prediction confidence obtained from the prediction vector. If the maximum prediction confidence exceeds a predefined threshold, the data record is classified as a member. Higher confidence indicates a stronger belief in the prediction made by the model.

Prediction Entropy-based MIA: This approach relies on prediction entropy, which measures the uncertainty or randomness associated with the model's predictions. If the prediction entropy falls below a preset threshold, indicating low uncertainty, the data record is classified as a member.

4.2.1.3 Factors Contributing to MIAs

Membership Inference Attacks (MIAs) succeed for several reasons.
Firstly, they exploit overfitting in machine learning models. Overfitting occurs when a model performs much better on its training data than on unseen test data, often due to its high complexity and limited training dataset size. Deep learning models like Deep Neural Networks (DNNs) are particularly prone to overfitting because they are overparameterized, allowing them to learn effectively from large datasets but also causing them to memorize noise or dataset specifics.
Additionally, ML models are trained repeatedly on the same instances, making them likely to memorize training data, and the finite size of the training dataset fails to represent the entire data distribution, hindering the model's ability to generalize to unseen data.
Secondly, the type of target model influences MIA success. The less sucebtible a models decision boundary is to a particular record the more resillent it is to MIAs. Lets look at decision tree models and a model that classifies emails as spam if the model is sensitive to certain keywords their presence could create new branches for certain keywords, if now an attacker has an email with such keyword the classification process can indicate the membership.

4.2.1.4 Property Inference Attack [18]

As is in the name, Property Inference attacks aim to extract general properties of the target ML model's training dataset that were not explicitly encoded as features or not correlated to the learning task [14]. An example of a property inference attack is the extraction of the ratio of women and men in a patient dataset. Such property inference attacks allow adversaries to gain insight on training data and could violate intellectual property of model owners, privacy and be the building block for more advanced attacks e.g. membership inference attacks [18]
Property Inference attacks can target either generative or discriminative models. Former try to learn underlying training data distribution and generate new data based on it later are mainly designed to solve classification problems [18]. One type of generative models are Generative Adversial Netowrks (GANs). We will focus on those in our further evaluations.

Threat Models

The goal of the adversary is to infer whether the target model's training dataset D_{target} has a certain property P.

Assumption 1: The adversary has a auxiliary dataset $D_{auxiliary}$ with the same distribution as D_{target}. The adversary leverages it to build local GANs and classifiers for the attack.

Assumption 2: The adversary has only access to the generator of the target model G_{target}. As seen before we differentiate Full black-box, adversary can just get generated samples blindly from the target black-box generator and partial black-box settings, adversary has no knowledge about the parameters of the target GAN but can construct the latent code

z to generate the corresponding sample from G_{target}. In a partial black box setting the adversary also has information about the architecture of the target GAN and the training algorithm.[18]

Attack Workflow

The attack can be divided in three phases. In a first phase the adversary queries G_{target} to produce synthetic samples. Next the adversary constructs a *property classifier* f_P tailored for classifying the previously generated samples with respect to the property that should be extracted e.g. if it's gender distribution of the samples in the training dataset, the property classifier predicts the gender of each sample. The classifier is trained with part of the auxiliary dataset disjoint from the underlying training dataset. In a last step the adversary predicts the property P_{infer} based on the output of the property classifier using a function over the prediction of the property classifier. Following graphic displays the 3 steps[18] :

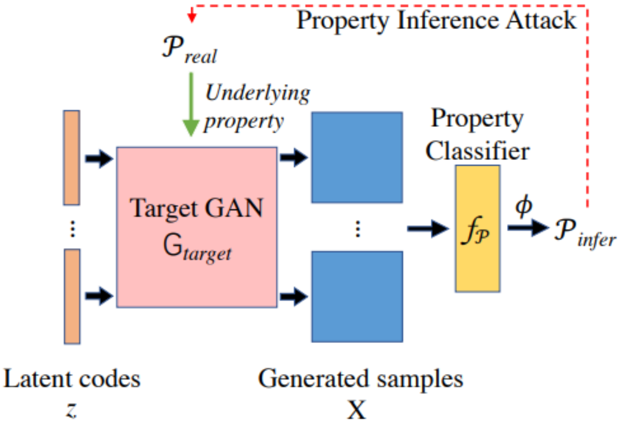


Figure 4.2: Workflow of property inference attack strategy [18]

4.2.2 Attacks aimed at the Model

4.2.2.1 Model Inversion attack [10]

Model Inversion attacks aim to reconstruct data samples from a target ML model i.e. they allow adversary to directly learn information about the training dataset.
For instance, in a facial recognition system, a Model Inversion adversary tries to learn the facial data of a victim whose data is used to train the model. Let's define Model Inversion attacks as follows:

$$M^{W,D_{aux}} \rightarrow \{\text{training samples}\}$$

where $D_{aux} \in \{D^N_{aux}, D^S_{aux}\}$ M^W: White-Box Model access D^N_{aux}: No auxiliary Dataset D^S_{aux}: Shadow auxiliary Dataset

M^W states that the adversary has white-box access of the target model, and D_{aux} is the auxiliary dataset, which is either empty or a shadow dataset. Following, we will look at two types of model inversion attacks:

- White-Box/No Auxiliary Model Inversion Attacks, which aim to reconstruct a representative sample for each class of the target model.
- White-Box/Auxiliary Model Inversion Attacks, which aim to synthesize the training dataset.

In White-Box no auxiliary Model Inversion Attack the adversary: first creates noise samples for each class of the target model e.g. for each identity create a facial image. Then the adversary feeds the samples to the model to get the probabilities of a class given a facial images. Then the adversary uses backpropagation over the available model parameters to optimize the input sample until the correseponding probability for a class exceeds a pre-set threshold. This optimized samples is then the attack output in our case the facial image the attacker connects to a given identity.

Now let's look at a second Model Inversion Attack type, here White-Box access and Shadow dataset are assumed. The shadow dataset is used to enhance the quality of the reconstructed samples by training a generative adversial network. First the GAN is trained with the shadow dataset Then the GAN generator is optimized meaning the noise; so that it generates samples that achiever higher posteriors on the target model Therefore here the GAN's input is optimized and enables the attacker to create high quality samples. As GAN can generate multiple samples from an input the attacker can create multiple e.g. facial images from a identity therefore raising the attacks chances of success.

4.2.2.2 Model extraction / stealing

We now turn to a type of inference attack called *model stealing* or *model extraction* attack. Fundamentally, this sort of attack is aimed at exactly what it's name implies, extracting a target model. However, some model extraction attack do not end there, they first extract the model and then use it to create what are called *transferable adversarial examples*. We touch on both of these cases below. As a final preliminary remark, only model extraction attacks where attackers have black-box access to a target model are sensible. If they had white-box access, there would be no point in mounting such an attack.

In any case, a model extraction attack needs to extract a model. More specifically, this means that an attacker tries to create their own *substitute model* which should accurately mimic the target model's outputs. For this they are interested in finding the hyperparameters the target model uses. An example of how this is achieved is provided by [7]. The algorithm they propose is roughly divided into two phases, a setup and a refinement phase. The setup is ran once, then multiple iterations of the refinement phase are run. Because these attacks rely on querying the target model, and because such queries are hardly ever free, Juuti et al. introduce a query budget, that an attacker does not want to exceed, to limit the costs for the attack. With target model t, substitute model s and query budget q, the algorithm as found at [7, p. 3] can be summed up as follows:

Setup - run once

1. Label initial training data by querying t.
2. Select architecture and hyperparameters for s.
3. Train s on labeled data.

Refinement - run until q is used up

1. Create synthetic samples from training data.
2. Label synthetic samples by querying t.
3. Add synthetic samples to labeled data.
4. Train s on labeled data.

What is used as initial data can vary, as implied by the taxonomy mentioned above, an attacker might have a partial dataset or a shadow dataset at their disposal, or have no dataset, which requires them to come up with their own initial training data. The setup is fairly straightforward, regarding the refinement phase; how the attacker creates synthetic data from the initial training data can vary, they might use another ML model or rely on different methods altogether.

If an attacker's goal is simply to steal a model to compete with the target model, this algorithm is all there is to the attack. As suggested in [10], the success of such an attack can be measured in how *precisely* the substitute model copies the target model's outputs.

There is however another motivation to perform model extraction, that adds another step to the attack. An attacker might use the substitute model they obtain, to find so-called *transferable adversarial examples*, which are discussed in [9]. An adversarial example is an input sample for which a model produces a wrong prediction. An adversarial example is transferable, if it can be transferred from one model to another, that is, if a model a produces a wrong output for an example input, we query a model b with that same example and it also produces a wrong output, then we have a transferable adversarial example. The idea of this avenue of model extraction is to extract a substitute model from the target model, as explained above, and then use the substitute model to find transferable adversarial examples, that will also lead to a wrong output if the target model is queried with them. Essentially this allows attackers to *deceive* ML models, which is significant as these models are tasked with financial transaction, facial recognition and automated driving among other things. A sign with a printed adversarial example that causes models used in automated driving to make wrong predictions could have potentially fatal consequences.

There is one distinction to be made with regards to transferable adversarial examples (TAE), which is also ofund in [9]; they can be *targeted* and *untargeted.* A targeted TAE has a specified wrong output it produces, to stick with the experiment found in [9], an targeted TAE for an image recognition model would cause the target model to output a specific, wrong, image description. Untargeted TAEs on the other hand do not specify anything beyond the output needing to be wrong. Liu et al. [9] specifically try to find TAEs that are minimally different from actual, correctly processed samples. In the case of images, this means that humans can still make out what is in the image, while the target model fails at the same task.

4.3 Countermeasures [6]

4.3.1 Against Membership inference

Two common defense mechanisms agianst Membership Inference Attacks are confindece score masking and regularization.

Confidence Score Masking: Confidence score masking aims to mitigate black-box MIAs on classification models by concealing true confidence scores returned by the target classifier. Three methods are employed: restricting the prediction vector to top-k confidence scores, providing only the prediction label, and adding noise to the prediction vector.

Regularization: Regularization techniques aim to reduce overfitting in ML models to mitigate MIAs. Methods such as L2-norm regularization, dropout, and adversarial regularization help models generalize better to test data and align their behaviors between training and test sets by the addition of a penalty term to the loss function, leading to an increase of bias (average difference between predicted and true value) and decrease of variance. However, while effective, these methods may struggle to balance membership privacy and model utility.

4.3.2 Against property inference [17]

One defense against property inference attacks was proposed by Stock et al. coined Property Unlearning. Following a graphic decpiting the property unlearning process

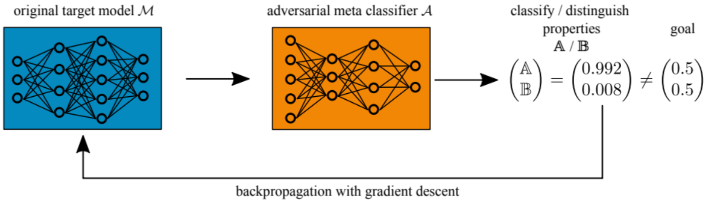
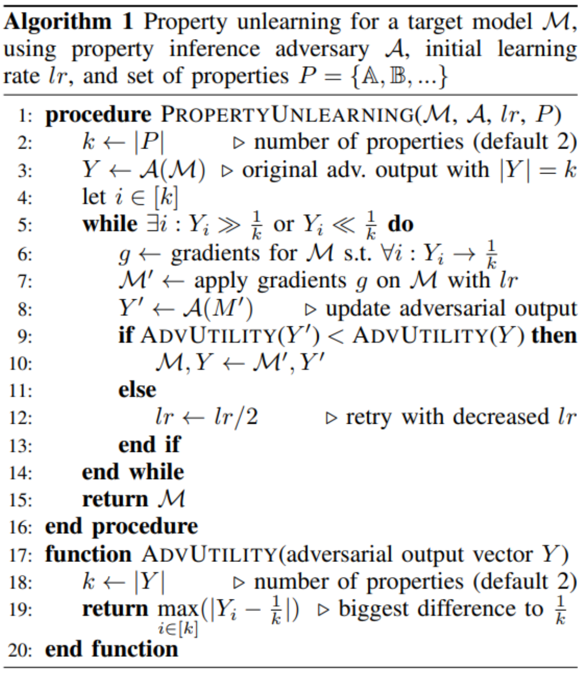

original target model $\mathcal{M}$
adversarial meta classifier $\mathcal{A}$
classify / distinguish properties $\mathbb{A}$ / $\mathbb{B}$
goal
$\begin{pmatrix}\mathbb{A}\\\mathbb{B}\end{pmatrix} = \begin{pmatrix}0.992\\0.008\end{pmatrix} \neq \begin{pmatrix}0.5\\0.5\end{pmatrix}$
backpropagation with gradient descent

Figure 4.3: Property Unlearning Figure
[17]

A prerequisite of the counter measure is the creation of an adversarial classifier. For it first for each property one auxiliary data set is created, then shadow models are trained for each property with the corresponding auxiliary data set. The auxiliary sets can be subsets of the training data set, thereby increasing the adversarial accuracy compared to outside adversaries. A second prerequisite is the full training of the target model. Backpropagation is then used to unlearn the property from the target model, hereby the parameters of the target model are modified by calculating and applying gradients with the goal to disable the adversary from extracting the property. The adversarial classifier output is in practice a vector whose components are the predicted probabilities for the present properties which sum up to 1. The algorithm stops when the adversarial classifier isn't anymore significantly more confident for one of the properties. Following the property unlearning algorithm.[17]



Algorithm 1 Property unlearning for a target model $\mathcal{M}$, using property inference adversary $\mathcal{A}$, initial learning rate lr, and set of properties $P = \{\mathbb{A}, \mathbb{B}, ...\}$

```
procedure PropertyUnlearning($\mathcal{M}$, $\mathcal{A}$, $lr$, $P$)
    $k \leftarrow |P|$    ▷ number of properties (default 2)
    $Y \leftarrow \mathcal{A}(\mathcal{M})$    ▷ original adv. output with $|Y| = k$
    let $i \in [k]$
    while $\exists i : Y_i \gg \frac{1}{k}$ or $Y_i \ll \frac{1}{k}$ do
        $g \leftarrow$ gradients for $\mathcal{M}$ s.t. $\forall i : Y_i \rightarrow \frac{1}{k}$
        $\mathcal{M}' \leftarrow$ apply gradients $g$ on $\mathcal{M}$ with $lr$
        $Y' \leftarrow \mathcal{A}(M')$    ▷ update adversarial output
        if AdvUtility($Y'$) < AdvUtility($Y$) then
            $\mathcal{M}, Y \leftarrow \mathcal{M}', Y'$
        else
            $lr \leftarrow lr/2$    ▷ retry with decreased $lr$
        end if
    end while
    return $\mathcal{M}$
end procedure
function AdvUtility(adversarial output vector $Y$)
    $k \leftarrow |Y|$    ▷ number of properties (default 2)
    return $\max_{i \in [k]}(|Y_i - \frac{1}{k}|)$    ▷ biggest difference to $\frac{1}{k}$
end function
```

Figure 4.4: Property Unlearning Algorithm [17]

4.3.3 Against model inversion [15]

In this section we will discuss two common countermeasures for model inversion attacks namely Homomorphic encryption and differential privacy (DP).

Homomorphic encryption: is one of several cryptographic techniques that can be utilized to perform ML training and testing. It enables computations to be carried out on ciphertext without ciphertext decryption or knowledge of any information of the plaintext. RSA and Elgamal are examples of multiplicatively homomorphic techniques. There the product of two ciphers is the encrypted product of two plaintexts. [15] The Faster CryptoNets framework, encrypted classification technique using neural networks by Chou et al. preserves the data privacy by using Hhmomorphic encryption, hereby revealing the output prediction only to the decryption key owner. [15]

Differential privacy (DP): is a perturbation approach, hereby random noise is added to the input data, model parameters during iteration of the trainig algorithm or the algorithm output to preserve data and model privacy [15]. Differential privacy was introduced by Dwork [1], and described in the context of privacy-preserving data mining where a trusted curator hold a private database. The curator responses to query issues by data analysts. Differential privacy guarantees that the query results are indistinguishable for two databases that differ only in one entry. Differential Privacy can be achieved by adding random noise to the dataset the common Gaussian mechanism computes a function on the dataset that adds random noise. The magnitude of the noise depends thereby from the change of the function output when a single individual data pointis added or removed, the so called global sensitivity.

4.3.4 Against model extraction / stealing

We discuss two proposed countermeasures against model extraction attacks, *prediction poisoning* [12], which aims to perturb attackers in training a substitute model, and *PRADA* [7] which relies on detecting model extraction attacks.

4.3.4.1 PRADA

Our discussion starts with PRADA (**Pr**otecting **A**gainst **D**NN Model Stealing **A**ttacks), as proposed by Juuti et al. [7]. PRADA's central mechanism is stated as follows:

> *"[...] PRADA's detection method is based on detecting deviations from a normal distribution in the distance between samples queried by a given client."* [7, p. 11]

As we have seen before, model extraction attacks rely on creating synthetic examples which are created during the refinement phase, based on the samples queried beforehand. PRADA's approach to detecting model extraction attacks rests on the intuition, that attackers *artificially control* the distance between these new synthetic samples, and the samples already queried, to maximize the amount of information gained by sending such queries. This means, that the *distances* between the queries such an attack involves, significantly deviate from a normal distribution.

The detection algorithm proposed by [7] is executed whenever a new query is received. It calculates the *minimum distance* of the incoming query, to any of the previous queries of that same client. That is, the distance between the incoming query, and the previous query which is closest to the incoming query is stored in a set. A so called *normality test* is then ran on the set of minimal distances, and checked against a threshold value, to determine whether an attack is taking place or not. Any defender would is required to provide this threshold value and a distance metric to indicate how far apart two samples are. This distance metric varies based on the sort of data at hand; it is domain-specific. The algorithm is outlined in detail at [7, p. 11].

Juuti et al. [7] test their defense against four proposed extraction attacks, two from external papers and two of their own. The results shows that picking too high of a threshold value results increases the false positive rate of the detection mechanism dramatically. Which is intuitive, since benign queries often have a close to normal distribution, rather than an exact normal distribution. The speed and accuracy of attack detection varies with the threshold value the defender chooses. There is one attack mentioned in the paper, which is not detected unless the threshold is set to a fairly high value, which resulted in a 0.1% false positive rate for PRADA. The reason the attack is not detected with a lower threshold, is that it uses a large step size in creating synthetic samples, which leads to bigger distances between these samples. The paper also mentions that setting such a high threshold is not always a viable strategy, since it may lead to an unacceptable false positive rate in certain domains.[7, pp. 12-14].

Overall, PRADA is a useful tool to detect many model extraction attacks. Having such detection mechanisms in place severely limits attackers' freedom in generating the samples they need for their attack, forcing them to ensure that the distance between their queries fits an inconspicuous distribution. There is a trade-off between setting the threshold to a high value, and keeping the false positive rate as low as possible, where a high threshold value ensures detection, but also increases the false positive rate at a certain point. A high false positive rate would pose a serious issue for the usability of any MLaaS application.

4.3.4.2 Prediction Poisoning

We now turn to prediction poisoning, a defense proposed by Orekondy et al.[12]. In contrast to PRADA, [12] does not rely on attack detection as they aim to avoid the assumptions regarding query distance distribution this entails. Instead, the proposed method of prediction poisoning works by altering the target models outputs with controlled perturbations, it is a perturbation-based defense. The trade-off in this approach is that a model's outputs should disrupt attacks as much as possible while remaining useful to benign users. Prediction poisoning relies on the insight, that an attacker does more with target model predictions than a benign user does; namely, attackers try to train a substitute model with these predictions. Orekondy et al. [12] find that earlier methods of perturbation-based defenses rely on hiding away uncertain, or overly precise predictions from the user, while giving them access to high confidence or sufficiently accurate predictions. For example, users might only get the first few decimals of a probability that there is a rabbit in a picture, which is alright for every-day use but is meant to inhibit training substitute models on such outputs. However, [12] claims that the impact this has on attacker success is often trivially small.

The approach of [12] is to target the optimization phase of the attacker's substitute model training. As discussed in the introduction, training a model involves optimizing it through backpropagation. This, in turn, relies on computing the gradient of an objective function. The proposed method of prediction poisoning aims to maximize the angular deviation of this gradient. That is, it aims to make sure that if an attacker tries to calculate this gradient based on the outputs of his substitute model, which is trained on the target models predictions, that this gradient deviates maximally from the actual gradient calculated for the target model.

Orekondy et al. find their method of defense to reduce attacker's performance by up to 53% on certain datasets, while reducing the defender's test accuracy only marginally.[12, p. 2]. They are able to hinder the accuracy of many of the tested attacks significantly, while retaining reasonable accuracy for the target model.

We discussed two kinds of defenses, a detection based and a perturbation based approach. The former relies on strong assumptions about the distribution of the distances between attackers queries. As we have touched upon in chapter 1.3.4, PRADA's detection based approach starts running into issues as attackers deviate from this assumption. An example is the attack not reliably detected because the distances between it's synthetic samples are bigger. PRADA was able to detect the attack with a very high threshold value i.e., by requiring queries to very closely fit a normal distribution in order to pass as benign. However, this high threshold value entails an increase in false positives, which can be unacceptable in many cases. The advantage of PRADA over perturbation based approaches, is that benign users receive the full, unaltered output of the model if they are not suspected of mounting an attack.

It seems to us, that prediction poisoning is a more promising approach to defending against model extraction. Firstly, perturbation-based defenses rely only on a very fundamental assumption, that attackers of model extraction attacks will train a model on the outputs of the target model. This seems a more secure assumption than the one enabling PRADA. Further, prediction poisoning as proposed by [12] seems to have a minimal impact on model usability. However, as users might use model outputs in different manners, it is imaginable that some use-cases require non-perturbed outputs.

It seems to us that detection and perturbation based approaches could conceivably be combined, where perturbation is only applied if queries are suspected to be adversarial. This might have the advantage of allowing some users access to the unaltered output of the model, while alleviating the problem of false positives for PRADA, as users deemed attackers would simply get slightly different outputs rather than being blocked from using the service altogether.

4.4 Conclusion

We have discussed several kinds of inference attack and some countermeasures proposed to deal with such attacks. In this section we touch upon each type briefly and weigh attacks against their countermeasures.

For Membership Inference Attacks we discussed confidence score masking, the concealing of true confidence scores, regularization techniques as L2-norm regularization increasing trading bias for variance, they both struggle to balance membership privacy with model utility.

The by Stock et al. coined Property Unlearning counter measure for property inference attacks makes use on an adversarial classifier to ensure that parameters of the target model are such that the output of the adversary is close to 1/k for all k entries of the output vector.

For Model Inversion Attacks we looked at one cryptographic technique called homomorphic encryption concealing the output prediction to the decryption key owner while allowing computations on the model, drawbacks can be high cost of encryption and at differential privacy a method adding random noise to input data so that the output of two by one element differing datasets is indistinguishable.

Regarding the feasibility of model extraction, it seems robust countermeasures exist which yield substantial results in hindering attackers from extracting the full functionality of a target model. The advantage a defender has is that any attackers need to mount a highly successful attack to be able to use their substitute model to compete with the target model. The substitute model would need to very accurately mimic the target model for such applications; the reviewed literature seems to suggest that while it might be possible to extract a substitute model with reasonable accuracy, the mentioned can defenses reliably prevent the extraction of critically accurate substitute model.

4.4.1 Compact Literature Overview

This subsection is intended to ease further research, it lists the reviewed literature as clearly and simply as possible and provides an at-a-glance comprehensive mapping from subject to literature.

- Deep Learning & Neural Networks: [8]
- Model extraction / stealing: [7, 12, 9]
- Membership Inference: [4]
- Model Inversion: [10]
- Property Inference: [18]
- Countermeasures
  - Model extraction / stealing
    - PRADA - attack detection [7]
    - Prediction poisoning [12]
  - Membership Inference
    - Confidence Score Masking [5]
    - Regularization [5]
  - Model Inversion
    - Homomorphic encryption [15]
    - Differential privacy [1, 10]
  - Property Inference
    - Property Unlearning [17]

Bibliography

[1] Cynthia Dwork. "Differential Privacy". In: *ICALP, Springer, pp.* (2006), pp. 1–12. DOI: https://doi.org/10.1007/11787006_1.

[2] Matthew Fredrikson et al. "Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing". In: San Diego, CA, USA: n 23rd USENIX Security Symposium (USENIX Security 14), 2014, pp. 17–32.

[3] Karan Ganju et al. "Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations". In: Toronto, ON, Canada: CCS '18: 2018 ACM SIGSAC Conference on Computer & Communications Security, Oct. 2018, pp. 619–633. DOI: 10.1145/3243734.3243834.

[4] Hongsheng Hu et al. *Membership Inference Attacks on Machine Learning: A Survey.* [cs.LG]. arXiv, 2022. URL: https://arxiv.org/abs/2103.07853.

[5] Hongsheng Hu et al. "Source Inference Attacks: Beyond Membership Inference Attacks in Federated Learning". In: *IEEE Transactions on Dependable and Secure Computing* (2023), pp. 1–18. DOI: 10.1109/TDSC.2023.3321565.

[6] Jinyuan Jia et al. "Memguard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples". In: London, United Kingdom: ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 259–274. DOI: 10.1145/3319535.3363201.

[7] Mika Juuti et al. "PRADA: Protecting Against DNN Model Stealing Attacks". In: Stockholm, Sweden: IEEE European Symposium on Security and Privacy (EuroS&P), 2019, pp. 512–527. DOI: 10.1109/EuroSP.2019.00044.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (May 2015), pp. 436–444. DOI: 10.1038/nature14539.

[9] Yanpei Liu et al. "Delving into Transferable Adversarial Examples and Black-box Attacks". In: International Conference on Representation Learning ICLR, 2017. DOI: 10.48550/arXiv.1611.02770.

[10] Yugeng Liu et al. "ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models". In: Boston, MA, USA: USENIX Security Symposium (USENIX Security 22), 2022, pp. 4525–4542.

[11] Lingjuan Lyu, Han Yu, and Qiang Yang. *Threats to Federated Learning: A Survey.* [cs.CR]. arXiv e-print, 2020. URL: https://arxiv.org/abs/2003.02133.

[12] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. "Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks". In: International Conference on Representation Learning (ICLR), 2020. DOI: 10.48550/arXiv.1906.10908.

[13] Anastasia Pustozerova and Rudolf Mayer. "Information leaks in federated learning". In: *Network and Distributed System Security Symposium* 10 (2020).

[14] Maria Rigaki and Sebastian Garcia. "A Survey of Privacy Attacks in Machine Learning". In: *ACM Computing Surveys* 56.4 (Nov. 2023), pp. 1557–7341. DOI: http://dx.doi.org/10.1145/3624010.

[15] Ahmed Shafee and Tasneem A. Awaad. "Privacy attacks against deep learning models and their countermeasures". In: *Journal of Systems Architecture* 114 (2021). Article 101940, pp. 1383–7621. DOI: https://doi.org/10.1016/j.sysarc.2020.101940.

[16] Reza Shokri, Marco Stronati, and Congzheng Song an Vitaly Shmatikov. "Membership inference attacks against machine learning models". In: San Jose, CA, USA: IEEE Symposium on Security and Privacy (S&P), 2017, pp. 3–18. DOI: 10.1109/SP.2017.41.

[17] Joshua Stock et al. "Lessons Learned: Defending Against Property Inference Attacks". In: *20th International Conference on Security and Cryptography*. SCITEPRESS - Science and Technology Publications, 2023, pp. 1–12. DOI: https://doi.org/10.5220/0012049200003555.

[18] Junhao Zhou et al. *Property Inference Attacks Against GANs*. [cs.CR]. 2021. URL: https://arxiv.org/abs/2111.07608.

Chapter 5

DNS Security and Privacy: The Landscape of Attacks and Mitigations

Andy Aidoo

Internet censors, opportunistic hackers, and security researchers have long engaged in a game of wits; this paper focuses on critical aspects of DNS seucrity and privacy in the domain name resolution process. Through the analysis of DNS protocols and Internet censorship methods, we develop crucial threat models that jeopardize the freedom of information on the Internet and its secure usage. We aim to provide guidance for domain name owners about secure DNS protocols to enhance integrity of DNS data. Our analysis includes protocols that cryptographically sign DNS data through public key cryptography. Furthermore, we aim to stress the significance of DNS transaction confidentiality for end-users. Our findings highlight the importance of wide-spread adoption of enhanced DNS protocols to not only improve individual security and privacy but also the Internet as a whole through herd security.

Contents

5.1 Introduction

The Domain Name System (DNS) represents a cornerstone of the Internet; its main purpose is translating human-readable domain names to Internet Protocol (IP) conforming addresses used in machine-to-machine communication. It precedes virtually every user-initiated network request and was crucial for the adoption of the Internet. Because of its critical role, it represents a significant target for cyberattacks, especially since DNS traffic is sent in clear text by default [23]. Attacks that manipulate DNS responses are numerous; opportunistic hackers target the domain name resolution process to hijack domain names from *i. a.*, government agencies and telecommunication companies [29]. Furthermore, state-sponsored Advanced Persistent Threats (APTs) frequently monitor DNS queries and prevent access to domains which are deemed to contain undesirable content [41, 25, 42]. Aforementioned DNS attacks can lead to severe consequences that result in service disruption, financial loss and data breaches. Over the years, numerous protocols such as the DNS Security Extension (DNSSEC) [1], DNS over TLS (DoT) [20], DNS over HTTPS (DoH) [17] and Oblivious DNS over HTTPS (ODoH) [28] have been proposed to enhance online security and privacy. DNSSEC aims to protect the authenticity and integrity of DNS responses, while the remaining protocols primarily prevent monitoring of users' Internet activities. Nonetheless, the adoption of these protocols remains inconsistent; many DNS queries remain unprotected. The fact that censors may employ protocol-based policy enforcement to hinder their adoption aggravates risks for users [43].

This paper seeks to provide a comprehensive overview of the DNS resolution process and dissects attack vectors through the introduction of two critical threat models. Furthermore, we evaluate mitigation strategies and their effectiveness in relation to Internet censorship. We aim to contribute to strengthening and potentially restoring freedom of information on the Internet by elucidating the proposed methods to assure the confidentiality and integrity DNS data. Moreover, we describe why aforementioned protocols should be used for benign DNS queries and not solely to circumvent censorship.

The remainder of the paper is structured as follows: Section 5.2 describes core concepts required to grasp the intricacies of improved DNS protocols. Section 5.3 focuses on securing DNS through means of security by means of obfuscation and cryptographic signatures. Subsequently, Section 5.4 concentrates on protocols that encrypt DNS queries and avoid potential cluster risks. Section 5.5 discusses the barriers to the adoption of integrity-focused and confidential domain name resolution before we finally conclude with a summary in Section 5.6.

5.2 Background

This Section provides essential background information about the Domain Name System (DNS) and introduces motivations for Internet censorship.

5.2.1 Domain Name System

Fundamentally, the domain name system is a decentralized database that maps memorable and human-readable domain names to machine-interpretable IP-addresses to *i. a.*, facilitate browsing the web through HTTP(S). It comprises of three major components:

5.2.1.1 Domain Name Space

The name space of a domain is represented in a tree-like structure and refers to the entire hierarchy of domain names under a specif top-level domain; its nodes and leaves contain a specified set of records that describe digital resources. Through queries said resources can be retrieved. Figure 5.1 presents a domain name space with *.com* as root of the tree; it is often referred to as Top-Level-Domain (TLD). The domain names of the nodes and leaves are the paths from the tree root to the respective nodes and leaves; they are also referred to as zones.

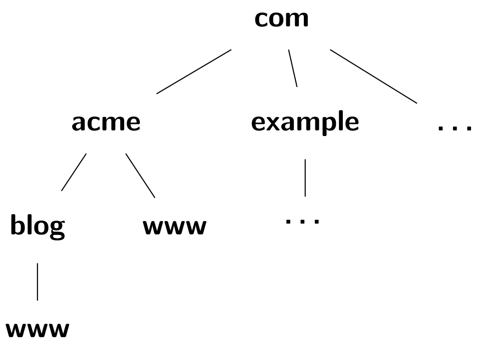


Figure 5.1: Illustration of a domain name space

There are various types of DNS records; Figure 5.2 illustrates two *A* type records that each encompass the associated host address for the domain names *acme.com* and *www.acme.com*. The third column indicates the records class, which in the example is the ARPA Internet system; lastly, the fourth column contains, in the case of *A*-records, the IP-address of the resources.

acme.com	A	IN	192.168.0.1
www.acme.com	A	IN	192.168.0.1

Figure 5.2: Example DNS Resource Record Set specifying a Domain's Host Addresses

5.2.1.2 Name Servers

Servers hosting a (subset of a) domain name space are dubbed *Name Servers*; their responsibilities include storing the hierarchy of domain trees and maintaining the values of DNS records. They typically denote the name servers of their subdomains in *NS*-Records and the IP address of the domain names they are responsible for in *A*-Records.

5.2.1.3 Resolvers

Finally, *Resolvers* are programs capable of extracting the DNS records from name servers in response to user requests. Whenever a user queries a domain name *e.g.*, *www.acme.com*, the (recursive) resolver proceeds to resolve it [12, 26].

5.2.2 DNS Resolution

Originally, DNS was designed without privacy in mind. Since its inception in the 1980s, DNS traffic has been transmitted in plain text. Despite the development of encrypted DNS protocols such as DoT and DoH, the majority of DNS queries are still sent in plain text [23]; Table 5.1 illustrates a simplified DNS query as observed on the wire.

DNS	Query:	Name: www.acme.com Type: A Class: IN
UDP	Source Port:	3141
	Destination Port:	53
IP	Source IP:	192.168.0.1
	Destination IP:	9.9.9.9

Table 5.1: Example DNS Query to resolve the Domain Name *www.acme.com*

Before users can access a web resource, they require the IP thereof. Usually, they only know the domain name, *e.g.*, www.acme.com and pass a query to a recursive resolver, as denoted by step ① in Figure 5.3; the recursive resolver may or may not be located within the network of the users. Provided the IP address of the domain is not available in its cache, the recursive resolver inquires the Name Server of the TLD by sending a query to one of the root Name Servers in step ② and expects a reply with the IP address of the TLD Name Server, as depicted in step ③. Subsequently, in step ④ the recursive resolver sends a Name Server request for the domain, in our example *acme.com* to the TLD Name Server which is responsible for domain name lookups of the *.com* domain name space. By processing the response in step ⑤, it can consult the Name Server responsible for subdomains of *acme.com*. These steps are recursively repeated (⑥ - ⑦) until a response includes the authoritative Name Server for the fully qualified domain name (FQDN). The recursive resolver can finally consult the authoritative Name Server for the IP in step ⑧ and receives the target IP address in step ⑨ which it then relays back to the users. This step ⑩ ends the DNS query for the recursive resolver. Users, on the other hand, are now capable of accessing the web resource via HTTP(s) or any other protocol built on top of the Internet Protocol (IP), as denoted in step ⑪.

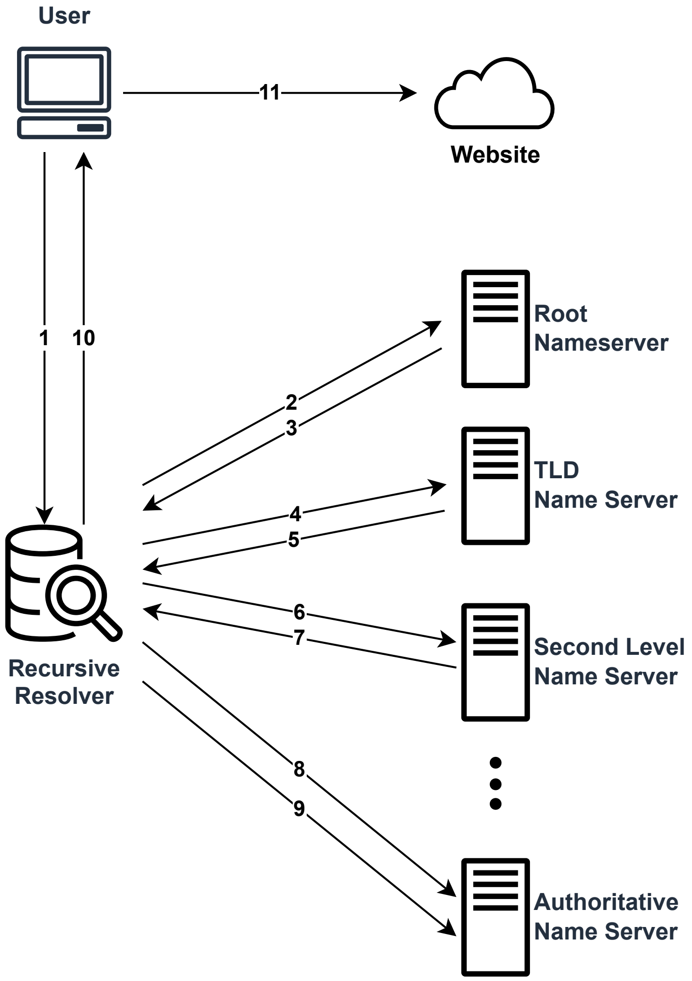


Figure 5.3: DNS Resolution Process

5.2.3 Internet Censorship

There are various forms and types of Internet censorship and the motivational sources are diverse; they include *i. a.*, political repression by targeting dissidents and human rights activists or critics of the state. Furthermore, Internet censorship may originate from religious authorities seeking to suppress the rise of ideas that are deemed heretical or sacrilegious [42]. Many of the Internet censorship techniques resemble man-in-the-middle (MITM) attacks where an threat actors monitor and alter network traffic. Censors can inspect DNS requests and may subsequently forge responses with error responses whenever the domain name of a forbidden domain is queried. Through so called deep packet inspection (DPI) censors are capable of inspecting the data of IP packets and perform censorship based on the packets' contents [32].

5.3 DNS Security

Since the domain name is sent in clear text, it allows adversaries to tamper with DNS lookup requests of their victims. This Section addresses security implications resulting from aforementioned DNS resolution process by developing a threat model and presenting mitigations to improve the security while looking up domain names.

5.3.1 Threat Model

We assume a threat model containing two parties: *Alice* who looks up domain names to retrieve IP addresses of web resources that she wants to access. Her adversary, *Eve*, aims to provoke access to fraudulent web resources. Eve is an off-path adversary; she possesses the capability of sending spoofed UDP packets to Alice.

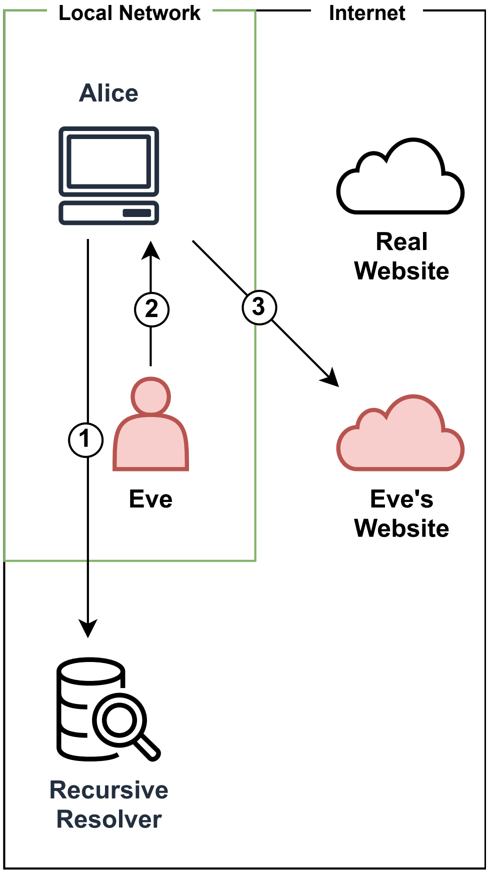


Figure 5.4: Rudimentary Example of DNS Poisoning

5.3.2 DNS Poisoning

In the first scenario, Eve is located within the same network as Alice with the fewest restrictions imposed by standard security measures. This scenario occurs when Alice connects to a public WIFI network that uses an obsolete WLAN security standard. Eve's aim is to serve a fraudulent web page whenever Alice visits the e-banking services of her bank to gather her account credentials. Figure 5.4 illustrates this simplified scenario that targets the victim machine, directly. Eve monitors traffic that is sent wirelessly and whenever she observes a DNS query from Alice (as shown in step ①), she notes the domain name in the query and if it corresponds to Alice's bank, she answers the request with a UDP packet containing the IP address of a server she controls and spoofs the IP address of the recursive resolver used by Alice as denoted in step ②. Subsequently, Alice unknowingly accesses Eve's website in step ③ instead of her bank's official website.

In a more realistic scenario (*cf.*, Figure 5.5), Eve is located outside of Alice's network but they share the same recursive resolver alongside other users. The interaction, again, starts with a DNS query to the recursive resolver from Alice, as shown in step ①. Before the recursive resolver receives an answer from an authoritative Name Server, Eve injects a fraudulent query response, step ③, such that the IP of her rogue server is returned in step ④ whenever Alice queries *e.g.*, the IP address her bank's e-banking service denoted by step ⑤. According to Radu and Hausding [34], around 50% of DNS queries are either resolved by Google's or Cloudflare's resolvers. To boost performance for the potentially millions of users that public resolvers serve IP addresses for, domain names are cached. According to Dagon *et al.* [11] a DNS query takes anywhere between 100 and 400 milliseconds, on average. This is the time frame in which Eve can send spoofed UDP packets to the publicly available recursive resolver. As illustrated by Daniel Kaminsky in 2008, the fact that DNS queries are sent via UDP, the lack of response verification and usage of caching can result in resolvers serving potentially millions of unsuspecting users rogue IP addresses in response to genuine domain name lookups. His attack features a specifically crafted payload in the query responses that next to the IP address of the domain name includes a self-declared authoritative name server for the parent domain; this would allow malicious actors to poison the cache for the IP addresses of TLD name servers [14]. To validate the authenticity of a query response most resolvers rely on non-cryptographic checks by using unpredictable values which are set by the resolvers themselves, *e.g.*, a 16-bit transaction ID and a randomly selected 16-bit source port in UDP requests that must be matched in the query response [21]. Dagon *et al.* [11] propose to further increase the entropy, *i.e.*, they introduce additional non-cryptographic verifiers to validate the authenticity of a query response by randomly capitalizing letters of the domain name.

Herzberg and Shulman [16], however, argue that the so-called *0x20* encoding does not sufficiently protect from DNS poisoning attacks, since the number of random case toggles is limited by the amount of characters in the domain name. The incompatibility of many resolvers further dampens the effectiveness of 0x20 encoding [7]. Moreover, using Kaminsky-style attacks on recursive resolvers means that an adversary only needs to beat the slightly harder guessing game once in order to poison a TLD and subsequently millions of domain names. Furthermore, Herzberg *et al.* [16] exploit the fact that resolvers often connect to the Internet using a network address translation (NAT) device. Essentially, source port randomisation from resolvers can completely be circumvented when poisoning the cache through UDP packets sent to the NAT device provided the source port randomisation of the NAT device can be circumvented. Finally, Man *et al.* [31] describe a novel approach to DNS poisoning by exploiting side-channels that target the operating systems of the servers hosting the resolvers. At the core, they exploit rate limits of the In-

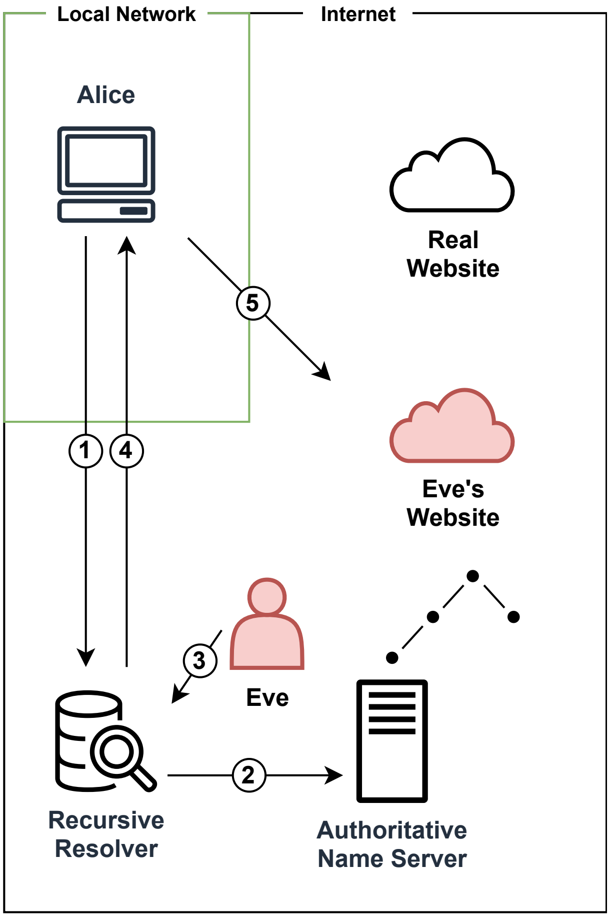


Figure 5.5: Example of DNS Cache Poisoning

ternet Control Message Protocol (ICMP) which are set between 200 and 1'000 per second depending on the operating system. Whenever a server receives an ICMP message on a closed port, *i.e.*, the port is not used by a resolver to lookup a domain name, it responds with a message indicated that the port is unreachable and thus allowing for port scanning of a server. By provoking delayed query responses, they increase the time available to identify the port used in a domain name lookup.

5.3.3 DNS Security Extension

With the aim of mitigating aforementioned vulnerabilities during the domain name resolution, the Internet Engineering Task Force (IETF) introduced DNSSEC almost three decades ago [1] and refined it in 2005 [36, 38, 37]. DNSSEC utilizes public key cryptography to provide integrity assurances for DNS data by augmenting the Name Servers with additional resource records.

5.3.3.1 Zone-Signing Keys

Before being able to validate the authenticity of a DNS response, the resource record sets (RRset), *cf.*, Figure 5.2, must be digitally signed by the zone-signing keys (ZSK) and the resulting digest stored as a RRset Signature (RRSIG) record in the Name Server. Furthermore, the public part of the ZSK pair must be added as a DNSKEY record in the Name Server to allow security-aware resolvers to validate the signature. A valid RRSIG guarantees that the correct DNSKEY has been provided in the DNS lookup response, *i.e.*, the response is cryptographically sound. Nonetheless, it does not necessarily prove authenticity of the response, as malicious actors can still provide all the required information themselves. Thus, further steps are required to ensure the authenticity of the received RRset.

5.3.3.2 Key-Signing Keys

As a second layer of security, the so-called Key-Signing keys (KSK) are introduced. Their responsibility encompasses providing means to authenticate the ZSK used in generating the RRSIG record. Analogously to the relationship between the RRset and the ZSK, the KSK produces a digest which is also returned during a DNS lookup. Security-aware resolvers are now able to validate the RRSIG alongside the authenticity of the ZSK that was used in its generation. Additionally, the ZSK is isolated and can be replaced straightforwardly in case it has been compromised. Figure 5.6 provides the architecture of a Name Server secured by DNSSEC.

5.3.3.3 Delegation Signer

Since the DNS is a hierarchical structure, individual domains depend on their parent domains. As a consequence, trust of a parent domain must be transferred to its child domains. Delegation Signer (DS) records serve this purpose; to configure a domain to use DNSSEC, operators must register a hash of their KSK with the parent domain which in turn adds it to the RRset and specifies it as a DS record. Security-aware resolvers retrieve the DS record of domains whose names they want to resolve and validate the KSK before authenticating the ZSK used to produce the RRSIG records. Provided that this chain of trust, as illustrated in Figure 5.7 resolves correctly, *i.e.*, each KSK's hash is present in the parents' DS records, the RRset provided by the authoritative Name Server of the domain can be trusted. Notably, changing a KSK requires an update in the parent's Name Server which depending on the configuration can take several hours to propagate [19].

5.4 DNS Privacy

Since the domain name is sent in clear text, it allows adversaries to monitor DNS lookup requests of their victims. This Section addresses privacy implications resulting from the DNS resolution process depicted in Section 5.2.2; we develop a threat model and present

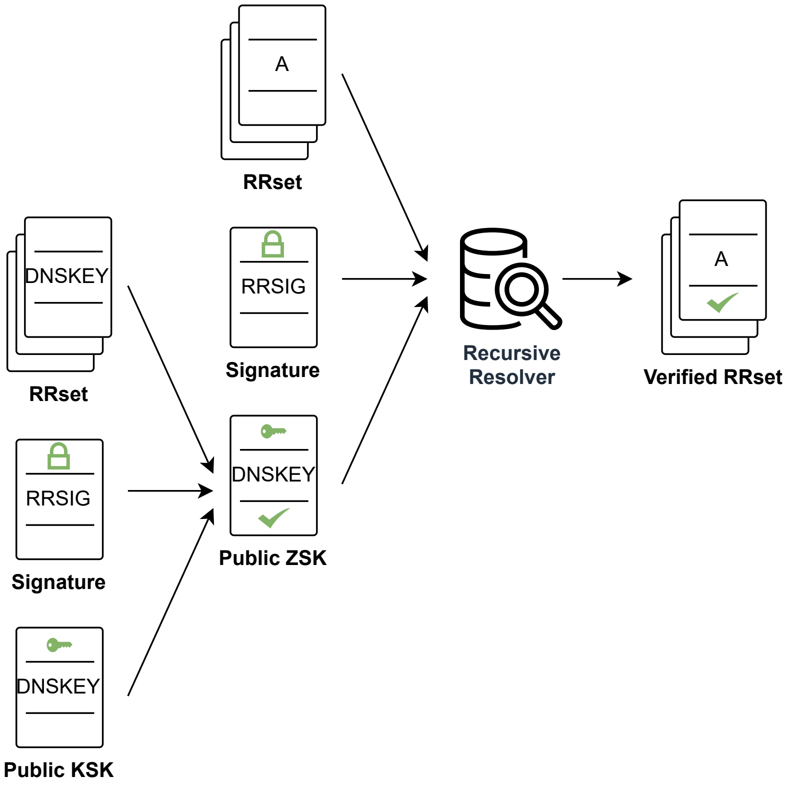


Figure 5.6: ZSK and KSK to assure the Authenticity of DNS Record Sets

mitigation techniques to preserve privacy and potentially evade censorship while looking up domain names.

5.4.1 Threat Model

To highlight the importance of privacy preserving domain name lookups, this threat model, again, contains two parties: *Alice* utilizes the Domain Name System to translate human-readable domain names to IP addresses which programs on her client device uses to communicate with web resources via communication protocols built on top of IP. This time, however, *Eve* is an on-path adversary with the aim of monitoring Alice's domain name queries to learn about her Internet activity and potentially censor access to, according to Eve, forbidden web resources. As an on-path adversary, she is capable of dropping packets originating from Alice's IP address and reply with spoofed IP addresses. Figure 5.8 depicts a visual representation of the threat model.

5.4.2 Traffic Monitoring

As previously discussed in Subsection 5.2.2, DNS queries are sent in clear text such that any requests made by Alice are visible to Eve. Since virtually every visit of a website is preceded by a domain name lookup, surveiling Alice through DNS requests becomes a simple yet effective way of monitoring her online activities. A rich body of literature

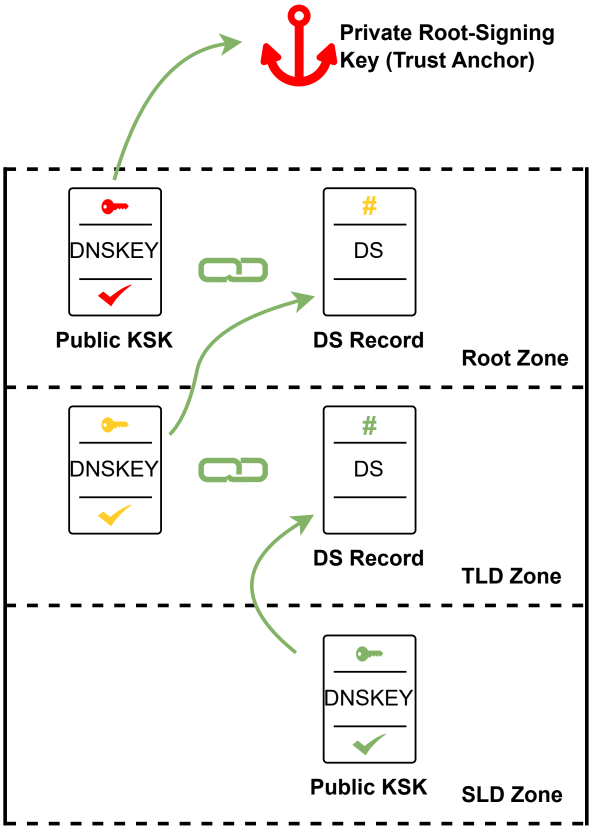


Figure 5.7: DNSSEC Chain of Trust

suggests various (non-) governmental organizations employ this approach when tracking and censoring Internet users [41, 27, 25, 44]; in some cases censorship which intended for a government's citizens results in collateral damages, *i.e.*, censorship spills over to Internet users of other countries [2, 22]. Restricting Internet usage based on domain names is especially attractive for censors, as domain names change less frequently than *e.g.*, IP addresses of the forbidden resources. To prevent access to such a resource, censors can drop the packet in step ① or ② and respond with the IP address of a block page as shown in Figure 5.8.

5.4.3 Privacy Preserving DNS

In this Subsection we delve into the realm of privacy-preserving DNS and examine protocols to encrypt DNS queries as well as further methods to anonymize DNS queries.

5.4.3.1 Encrypted DNS

Like any other traffic, packets may travel on various paths through the network; thus, adversaries are best served by monitoring the connection between the client and the recursive resolver instead of the path between the recursive resolver and an authoritative Name Server [5], as noted in ① and ② of Figure 5.8. As a mitigation to this threat, *i.*

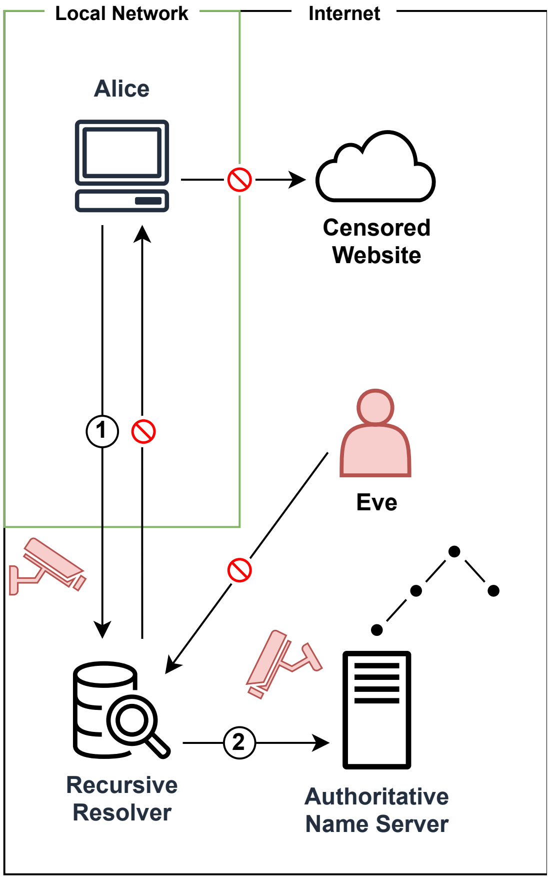


Figure 5.8: DNS Privacy Threat Model with an On-Path Adversary

a., DoT [20] and DoH [17] have been developed. To utilize DoT, a privacy-conscientious client must establish a TCP connection on port 853 of the DoT-resolver. Once the TCP handshake has concluded, the client can send encrypted DNS queries to the DoT-resolver. The payload of the DNS query does not differ from the regular DNS query [20].

DoH, on the other hand, deviates from the default DNS query as the client can sends DNS queries through HTTP-*GET* or HTTP-*POST* requests to the well-known port 443 and therefore making DNS queries practically indistinguishable from regular HTTPS traffic. Within the HTTP request header the content type should specify *application/dns-message* to maximize cache friendliness. HTTP-*GET* requests must include the domain name in a query parameter called *dns*, whereas the body of an HTTP-*POST* request includes the domain name which the client is interested in [17].

The change from UDP to TCP introduces additional latency when the session is initiated and results in an increase of required resources caused by the request encryption and decryption. Nevertheless, Böttger *et al.* [6] and Hounsel *et al.* [18] both conclude that the computational overhead required in resolving DNS queries through DoT and DoH, respectively, are negligible. The benefits of better privacy guarantees do not result in perceivable increased page loading times.

Querying DNS names through DoH or DoT has the beneficial side-effect that censors cannot as easily block access to censored domains; Jin *et al.* [25] discovered that switching to DoH or DoT resulted in the circumvention of most state-imposed censorship. Despite these promising results, Siby *et al.* [40] demonstrate how ciphertext analysis permits highly accurate identification of domains through DNS fingerprinting. Initially, they collect network traces that are associated with a visit of a website, *e.g.*, accessing the website of the British Broadcasting Corporation results not only in the domain name lookup of *bbc.com*, but also 51 further requests to various subdomains and third-party domains. The amount, timings and sizes of DNS queries and responses are highly indicative of the website which is visited and would allow a censorship adversary to identify access to undesired websites. As a countermeasure they suggest adding padding to the DNS responses in order to eliminate packet size information, analogously to the repacketization strategy employed in Tor [30].

5.4.3.2 Anonymous DNS

As highlighted by Radu *et al.* [34] the majority of DNS queries world-wide are resolved by only a small number of large service providers. Furthermore, the fact that authoritative government frequently mandate ISPs to censor undesirable content on the Internet [22] emphasizes the need for further advances in anonymization in DNS queries. A rudimentary approach employs so-called range queries where apart from the desired domain, further domain names are added to the query to introduce noise which hides user activity [45]. Next to significantly increased bandwidth consumption, this approach is accompanied with serious implementation challenges while not providing sufficient privacy guarantees, as the users' queries are still deducible [8]. Contrarily, the experimental protocol ODoH has been developed to allow DNS queries where no single server is aware of the client IP address and DNS query at once while preserving the confidentiality and integrity of

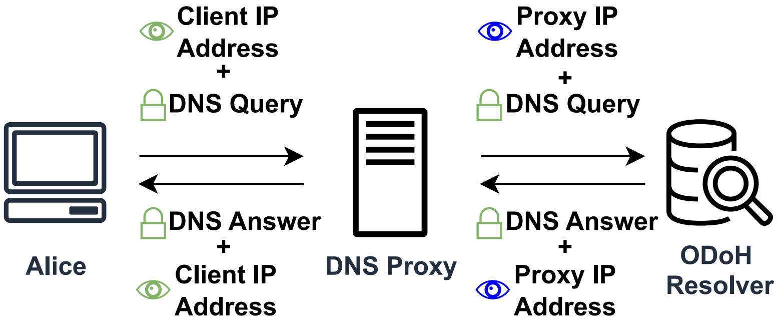


Figure 5.9: DNS Query through Oblivious DNS over HTTPS

the query contents. The ODoH protocol differs from DoH by introducing a proxy that functions akin to a NAT device; *i.e.*, the proxy maintains a list of IP addresses it has forwarded queries for and relays the corresponding response to the right IP address. To satisfy performance and integrity constraints, a hybrid public key encryption (HPKE) scheme is used. As a first step, clients retrieve the public key of a ODoH resolver to encrypt a freshly generated shared secret. The shared secret is then used to encrypt the payload which in this case is the DNS query. This approach guarantees that the only party capable of deciphering the shared secret is the owner of the key pair used in the encryption [3]. Clients can query an ODoH server by sending HTTP-*POST* requests to a DNS proxy. In the request the client must specify the target host and path in the query parameters; the encrypted DNS query resides in the body of the request. Target host refers to the host name of the ODoH server, while target path describes the endpoint at which the DNS query can be resolved. As response the ODoH server sends the encrypted DNS query answer to the proxy, which in turn forwards it to the DNS client resulting in an anonymous DNS lookup [28]. Figure 5.9 illustrates the DNS query flow.

5.5 Discussion

For the threat models, we assumes rational adversaries; *i.e.*, the costs associated with DNS poisoning or censorship must be lower than the financial or political gains. For instance, the importance and economic added value of GitHub presents a dilemma for authoritarian governments. Without the availability of this global code repository, technology companies are severely handicapped in producing state-of-the-art technology products. As a consequence, GitHub is still available in authoritarian regimes despite being used as a place to host anti-government newsletters and to collectively organize protests against such governments as illustrated by the popular repository 996.ICU [1] [13]. This Section will discuss the viability of DNSSEC and encrypted DNS against aforementioned threat models in the Subsections 5.4.1 and 5.4.1.

[1] `https://github.com/996icu/996.ICU`

5.5.1 DNSSEC

DNSSEC was introduced in 2005 and mainly addresses the integrity of DNS query responses [36]; *i.e.*, it prevents adversaries from injecting fraudulent DNS responses. Despite its primary focus on security and not on privacy (DNS request are still sent in clear text), it increases the barrier for censorship. In this context, DNSSEC provides apt mitigation where censorship is based on the exploitation of race conditions in the DNS protocol. Rather than accepting the first DNS response, which in authoritarian regimes typically corresponds to a *Domain Not Found*-error or the IP address of a server hosting a block page [33], the recursive resolver that is queried by the users awaits a legitimate response. Consequently, censors can no longer rely on aforementioned straightforward censorship approach. To successfully restrict access to blacklisted domains, censors would either have to possess the capability of dropping UDP packets containing DNS queries for blocked domains or employ IP address-based censorship. The prevalent usage of content delivery networks (CDNs) to decrease latency when accessing websites often results in numerous domains being hosted from the same server [10]; *i.e.*, an IP based-blocking approach may lack sufficient granularity and censors cannot rely on it, lest they incur collateral damage.

The adoption rate of DNSSEC varies considerably and requires a nuanced examination; Table 5.2 lists the most popular TLDs alongside the number of second level domains, the number of domains employing DNSSEC and finally, the resulting adoption rate. The TLD *com* shows an adoption rate of 5% and all other but the *ch* and *se* TLD disclose an adoption rate roughly 70%. However, signing DNS records only corresponds to half of the activities involved in using DNSSEC. Clients must validate the signatures in order to reap the benefits of DNSSEC. Consequently, a meaningful adoption rate must also take client validation and the long tailed distribution of the queries themselves into consideration; *i.e.*, the majority of DNS queries contain the names of a small minority of domain names. According to Huston's analysis of DNSSEC usage by Cloudflare's DNS resolver, only 1% of DNS traffic actually utilizes DNSSEC; additional operational costs and risks that are associated with employing DNSSEC further function as barriers slowing down its adoption [24]. One such risk is the increased query response that might exceed the maximum transmission unit due to the incorporation of digital signature. van Rijswijk-Deij discovered that a considerable amount of DNS resolvers cannot cope with fragmented DNS responses [35]. Finally, DNSSEC typically incorporates a mechanism for authenticated denial of existence for domains; this mechanism essentially presents proof to a resolver that a given domain name does not exists, which in a cryptographically verifiable way. Withing the RRSet of a domain, *e.g.*, acme.com, the record type *NSEC* links to a further domain, *e.g.*, a.acme.com, typically in alphabetical order. In this example, the NSEC record points to the subdomain c.acme.com; this allows a resolver to conclude that the subdomain b.acme.com does not exists [15]. As a consequence, all domain names can be discovered through repeated querying of the NSEC records, which can be in direct conflict with policies, *e.g.*, when a "confidential" subdomain is used as a employee login page.

Based on the low adoption rate, aforementioned barriers for adoption and the increased complexity resulting from key management for digital signatures, the primitive approach of injecting false DNS responses will likely remain a resourceful approach to Internet censorship. Furthermore, such a low adoption rate allows censors to prohibit and block

TLD	Domains (mio.)	DNSSEC (mio.)	Adoption Rate
com	156	7.1	5%
net	12	0.7	5%
org	10	0.6	6%
info	3.6	0.1	4%
ch	2.5	1.9	73%
se	1.4	1.0	72%
other	5.0	0.4	7%
Total	**192**	**11.8**	**6%**

Table 5.2: DNSSEC Adoption Rate by TLD
*Derived from `https://www.statdns.com/` Report of May 2024

queries aimed at retrieving DS resource records without incurring significant collateral damages.

5.5.2 Privacy Preserving DNS

Censorship of DNS queries occurs in democracies and authoritarian governments, alike. Since DNS queries precede virtually any Internet traffic and are usually transmitted in plaintext, DNS manipulation is often employed as means for censorship. To combat traffic monitoring and ultimately Internet censorship, methods for privacy preserving DNS querying can be leveraged. Rather than blocking individual IP addresses, the more memorable and consistent domain names are censored. Provided a DNS server is outside the control of the censors and supports the protocols DoT, DoT or ODoH, censorship can often be evaded. Jin *et al.* [25] discovered that a large portion of censored domains become reachable through DoT or DoH in China Denmark and Portugal. Since DoT targets port 853 of the DNS resolver, such traffic is easily identifiable and therefore subject to interference, as observed in Iran. Basso *et al.* [43] report that the majority of DoT resolvers are blocked by Iranian ISPs, however, hardly any DNS requests over HTTPS have been blocked during their experiments. Nonetheless, website identifiers, such as domain names not only leak information during DNS queries, but also when traffic is sent using HTTPS; the *Host* header in HTTP requests usually includes the domain name in cleartext and can thus easily be censored even if encrypted DNS is used. As mitigation, *TLS 1.3* introduced the encryption of aforementioned website identifier, which prevents censorship based on domain names entirely, provided that an encrypted DNS protocol is used in conjunction. However, the success of this protocol depends on its popularity; *i.e.*, the protocol should be used predominately on the Internet for benign traffic and not only to circumvent Internet restrictions lest censors can block this protocol without incurring substantial collateral damage [9]. Initial signs of resistance from censors has been observed in 2020, where users reported censorship of connections using TLS 1.3 in China [4]; furthermore, the Russian government has published an amendment to its technology laws in order to ban Internet protocols that hinder its effectiveness in surveillance and censorship due to encrypted website identifiers [39].

5.6 Summary

The interplay between adversaries and defenders in the resolution of domain names is characterized by a cat-and-mouse game; the former role is assumed by Internet users, while the counterparties consist of state-sponsored advanced persistent threats (APT), web tracking giants and opportunistic hackers. When the domain name system was designed, security and privacy were not perceived as a priority. As the usage of the Internet relies on resolving domain names to IP addresses, it presents a convenient attack vector to exploit. Security researcher Daniel J. Bernstein warned of DNS hijacking vulnerabilities in the early 2000s. Since then, the reliance on the Internet has gradually increase and entire business models and economies depend on correct domain name resolution. Furthermore, APTs utilize the optimistic approach of DNS to censors the Internet usage of a countries citizens, whereas opportunistic hackers hijack domain names with the goal of gathering intelligence and user credentials. While numerous protocols like DNSSEC and end-to-end encrypted DNS have been developed to address and mitigate risks, their adoption has been low for years. Defenders are at an inherent disadvantage, because they utilize a potentially compromised and adversary-controlled network; as a consequence adversaries can refuse to process requests using secure and privacy-preserving protocols. Nonetheless, provided that the world-wide adoption of protocols like DNSSEC and ODoH increases, the abilities of adversaries to eavesdrop and restrict Internet usage disappears. Thus, unrestricted and secure Internet usage relies on herd security where the majority of network participants should§ utilize secure and privacy-preserving protocols, as blocking thereof would result in substantial collateral damage.

Bibliography

[1] Donald E. Eastlake 3rd and Charles W. Kaufman. *Domain Name System Security Extensions*. RFC 2065. Accessed: 2024-04-18. Jan. 1997. DOI: 10.17487/RFC2065. URL: https://www.rfc-editor.org/info/rfc2065.

[2] Anonymous. "The collateral damage of internet censorship by DNS injection". In: *SIGCOMM Comput. Commun. Rev.* 42.3 (June 2012), pp. 21–27. ISSN: 0146-4833. DOI: 10.1145/2317307.2317311. URL: https://doi.org/10.1145/2317307.2317311.

[3] Richard Barnes et al. *Hybrid Public Key Encryption*. RFC 9180. Accessed: 2024-05-15. Feb. 2022. DOI: 10.17487/RFC9180. URL: https://www.rfc-editor.org/info/rfc9180.

[4] Kevin Bock et al. *Exposing and circumventing China's censorship of esni*. Aug. 2020. URL: https://geneva.cs.umd.edu/posts/china-censors-esni/esni/.

[5] Stéphane Bortzmeyer. *DNS Privacy Considerations*. RFC 7626. Accessed: 2024-05-15. Aug. 2015. DOI: 10.17487/RFC7626. URL: https://www.rfc-editor.org/info/rfc7626.

[6] Timm Böttger et al. "An Empirical Study of the Cost of DNS-over-HTTPS". In: *Proceedings of the Internet Measurement Conference*. IMC '19. Amsterdam, Netherlands: Association for Computing Machinery, 2019, pp. 15–21. ISBN: 9781450369480. DOI: 10.1145/3355369.3355575. URL: https://doi.org/10.1145/3355369.3355575.

[7] *Case randomization recently disabled?* Accessed: 2024-04-19. Feb. 2019. URL: https://community.cloudflare.com/t/case-randomization-recently-disabled/61376.

[8] Sergio Castillo-Perez and Joaquin Garcia-Alfaro. "Evaluation of two privacy-preserving protocols for the DNS". In: *2009 Sixth International Conference on Information Technology: New Generations*. IEEE. 2009, pp. 411–416.

[9] Zimo Chai, Amirhossein Ghafari, and Amir Houmansadr. "On the Importance of Encrypted-SNI (ESNI) to Censorship Circumvention". In: *9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19)*. 2019.

[10] Joachim Charzinski. "Traffic properties, client side cachability and CDN usage of popular web sites". In: *International GI/ITG Conference on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*. Springer. 2010, pp. 136–150.

[11] David Dagon et al. "Increased DNS forgery resistance through 0x20-bit encoding: security via leet queries". In: *Proceedings of the 15th ACM conference on Computer and communications security*. 2008, pp. 211–222.

[12] *Domain names: Implementation specification.* RFC 883. Nov. 1983. DOI: 10.17487/RFC0883. URL: https://www.rfc-editor.org/info/rfc883.

[13] Emily Feng. *GitHub has become a haven for China's censored internet users.* Accessed: 2024-05-25. Apr. 2019. URL: https://www.npr.org/2019/04/10/709490855/github-has-become-a-haven-for-chinas-censored-internet-users.

[14] Steve Friedl. *Steve Friedl's unixwiz.net tech tips.* Accessed: 2024-05-08. Aug. 2008. URL: http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html.

[15] R. (Miek) Gieben and Matthijs Mekking. *Authenticated Denial of Existence in the DNS.* RFC 7129. Accessed: 2024-05-27. Feb. 2014. DOI: 10.17487/RFC7129. URL: https://www.rfc-editor.org/info/rfc7129.

[16] Amir Herzberg and Haya Shulman. "Security of patched DNS". In: *Computer Security–ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10-12, 2012. Proceedings 17.* Springer. 2012, pp. 271–288.

[17] Paul E. Hoffman and Patrick McManus. *DNS Queries over HTTPS (DoH).* RFC 8484. Accessed: 2024-05-15. Oct. 2018. DOI: 10.17487/RFC8484. URL: https://www.rfc-editor.org/info/rfc8484.

[18] Austin Hounsel et al. "Analyzing the costs (and benefits) of DNS, DoT, and DoH for the modern web". In: *Proceedings of the applied networking research workshop.* 2019, pp. 20–22.

[19] *How DNSSEC Works.* Accessed: 2024-04-18. URL: https://www.cloudflare.com/dns/dnssec/how-dnssec-works/.

[20] Zi Hu et al. *Specification for DNS over Transport Layer Security (TLS).* RFC 7858. Accessed: 2024-05-15. May 2016. DOI: 10.17487/RFC7858. URL: https://www.rfc-editor.org/info/rfc7858.

[21] Bert Hubert and Remco Mook. *Measures for Making DNS More Resilient against Forged Answers.* RFC 5452. Accessed: 2024-05-21. Jan. 2009. DOI: 10.17487/RFC5452. URL: https://www.rfc-editor.org/info/rfc5452.

[22] Philip Hunter. "Pakistan YouTube block exposes fundamental Internet security weakness: Concern that Pakistani action affected YouTube access elsewhere in world". In: *Computer Fraud & Security* 2008.4 (2008), pp. 10–11. ISSN: 1361-3723. DOI: https://doi.org/10.1016/S1361-3723(08)70065-4. URL: https://www.sciencedirect.com/science/article/pii/S1361372308700654.

[23] Geoff Huston. *Doh, dot, and Plain Old Dns.* Sept. 2022. URL: https://blog.apnic.net/2022/09/02/doh-dot-and-plain-old-dns/.

[24] Geoff Huston. *Measuting the use of DNSSEC.* Accessed: 2024-05-12. Sept. 2023. URL: https://blog.apnic.net/2023/09/18/measuring-the-use-of-dnssec/.

[25] Lin Jin et al. "Understanding the impact of encrypted DNS on internet censorship". In: *Proceedings of the Web Conference 2021.* 2021, pp. 484–495.

[26] Aminollah Khormali et al. "Domain name system security and privacy: A contemporary survey". In: *Computer Networks* 185 (2021), p. 107699. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2020.107699. URL: https://www.sciencedirect.com/science/article/pii/S1389128620313001.

[27] Dae Wook Kim and Junjie Zhang. "You are how you query: Deriving behavioral fingerprints from DNS traffic". In: *Security and Privacy in Communication Networks: 11th EAI International Conference, SecureComm 2015, Dallas, TX, USA, October 26-29, 2015, Proceedings 11.* Springer. 2015, pp. 348–366.

[28] Eric Kinnear et al. *Oblivious DNS over HTTPS*. RFC 9230. Accessed: 2024-05-15. June 2022. DOI: 10.17487/RFC9230. URL: https://www.rfc-editor.org/info/rfc9230.

[29] Brian Krebs. *A Deep Dive on the Recent Widespread DNS Hijacking Attacks*. Accessed: 2024-05-30. Feb. 2019. URL: https://krebsonsecurity.com/2019/02/a-deep-dive-on-the-recent-widespread-dns-hijacking-attacks/.

[30] Zhen Ling et al. "Equal-sized cells mean equal-sized packets in Tor?" In: *2011 IEEE International Conference on Communications (ICC)*. IEEE. 2011, pp. 1–6.

[31] Keyu Man et al. "Dns cache poisoning attack reloaded: Revolutions with side channels". In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 1337–1350.

[32] Arian Akhavan Niaki et al. "ICLab: A global, longitudinal internet censorship measurement platform". In: *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2020, pp. 135–151.

[33] Sadia Nourin et al. "Measuring and Evading Turkmenistan's Internet Censorship: A Case Study in Large-Scale Measurements of a Low-Penetration Country". In: *Proceedings of the ACM Web Conference 2023*. WWW '23. <conf-loc>, <city>Austin</city>, <state>TX</state>, <country>USA</country>, </conf-loc>: Association for Computing Machinery, 2023, pp. 1969–1979. ISBN: 9781450394161. DOI: 10.1145/3543507.3583189. URL: https://doi.org/10.1145/3543507.3583189.

[34] Roxana Radu and Michael Hausding. "Consolidation in the DNS resolver market–how much, how fast, how dangerous?" In: *Journal of Cyber Policy* 5.1 (2020), pp. 46–64.

[35] Roland Martijn van Rijswijk-Deij. *Improving DNS security: a measurement-based approach*. University of Twente, 2017.

[36] Scott Rose et al. *DNS Security Introduction and Requirements*. RFC 4033. Accessed: 2024-04-18. Mar. 2005. DOI: 10.17487/RFC4033. URL: https://www.rfc-editor.org/info/rfc4033.

[37] Scott Rose et al. *Protocol Modifications for the DNS Security Extensions*. RFC 4035. Accessed: 2024-04-18. Mar. 2005. DOI: 10.17487/RFC4035. URL: https://www.rfc-editor.org/info/rfc4035.

[38] Scott Rose et al. *Resource Records for the DNS Security Extensions*. RFC 4034. Accessed: 2024-04-18. Mar. 2005. DOI: 10.17487/RFC4034. URL: https://www.rfc-editor.org/info/rfc4034.

[39] *Russia wants to ban the use of secure protocols such as TLS 1.3, DoH, DoT, ESNI*. Accessed: 2024-05-25. Sept. 2020. URL: https://www.zdnet.com/article/russia-wants-to-ban-the-use-of-secure-protocols-such-as-tls-1-3-doh-dot-esni/.

[40] Sandra Siby et al. "Encrypted DNS → privacy? A traffic analysis perspective". In: *arXiv preprint arXiv:1906.09682* (2019).

[41] Matthias Wachs, Martin Schanzenbach, and Christian Grothoff. "On the feasibility of a censorship resistant decentralized name system". In: *International Symposium on Foundations and Practice of Security*. Springer. 2013, pp. 19–30.

[42] Barney Warf. "Geographies of global Internet censorship". In: *GeoJournal* 76 (2011), pp. 1–23.

[43] Maria Xynou et al. *Iran blocks social media, app stores and encrypted DNS amid Mahsa Amini protests*. Accessed: 2024-05-25. Sept. 2022. URL: `https://ooni.org/post/2022-iran-blocks-social-media-mahsa-amini-protests`.

[44] Zhiwei Yan and Jong-Hyouk Lee. "The road to DNS privacy". In: *Future Generation Computer Systems* 112 (2020), pp. 604–611.

[45] Fangming Zhao, Yoshiaki Hori, and Kouichi Sakurai. "Two-servers PIR based DNS query scheme with privacy-preserving". In: *The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007)*. IEEE. 2007, pp. 299–302.

Chapter 6

Impact of Heterogeneous IoT Networks in Case of Smart Homes on Privacy

Emanuel Frank, Matthias Mylaeus

Abstract

Many Internet of Things (IoT) technologies are engrained within smart homes, creating a complicated interplay of convenience, efficiency, and security. This chapter examines the multiple implications of diverse IoT networks on privacy within the smart home ecosystem. In this context, present-day technologies are presented with their progressions and accompanying difficulties, especially regarding smart lock systems, environmental sustainability, health, and access management solutions. Furthermore, light is shed on the conflict between strengthening security and maintaining one's privacy.

The vulnerability exposed by less recognized IoT device manufacturers, alongside risks arising from commonly used security gadgets, are discussed. Moreover, this chapter discusses energy consumption, the sustainability implications of smart home technologies, and the probable health consequences of over-reliance on digital assistants.

Apart from that, the legal frameworks are analyzed, such as the General Data Protection Regulation (GDPR), which aims to protect customers against privacy breaches, including the recently updated new Federal Act on Data Protection (nFADP) from Switzerland. Since new regulations are in power, this is currently an important topic. It is crucial that all participants involved in smart homes, as well as customers and manufacturers, understand the current legal landscape. This chapter has identified some critical areas for potential future research, emphasizing their significance.

Contents

6.1 Introduction

With the help of the IoT, electronic devices and sensors can communicate with each other over the internet [20]. This interconnectivity aims to enhance automation in smart homes, improve efficiency, and provide real-time data and insights, ultimately leading to smarter decision-making and enhanced convenience in various aspects of life. With the development of new possibilities for integrating and extending customizable features, the number of such devices is increasing daily. Meanwhile, IoT is interleaved in a lot of different domains [21], spanning from industry [45] over entertainment all the way to healthcare [17]. However, as much as the IoT provides support, this growth and increased use also bring several challenges. Not only does the communication load become greater, but concerns about privacy and security arise more evidently [20].

In particular, Smart Home Systems (SHS), including efficient, effective, and reliable automation systems, are considered a remarkable transformation in daily life routines [47], and therefore present the main focus of this chapter. Due to their affordability and ease of use, smart devices are increasingly more popular [24]. This chapter discusses various protocols such as Ultra-Wideband (UWB) [1, 48], Multi-Access Edge Computing (MEC) [23, 31] and the 128-Bit AES-CCM Algorithm [10]. The importance of these protocols lies in their ability to enhance the security of IoT systems, which directly influences the protection of personal privacy.

Achieving sustainability in SHS is an additional challenge. Sustainable technology practices aim to reduce energy consumption and environmental footprints. Heterogeneous systems involving diverse and distributed computing resources (e.g., IoT devices and cloud services) can increase energy use. Ensuring that these systems are energy-efficient while maintaining robust privacy protections is difficult. Advancements for reducing energy use and enhancing efficiency across not only homes but also cities are significant [19]. Possible integrations for using renewable energy are essential for a smaller environmental footprint and, at the same time, enable it to shape the future positively, evidently leading to an improved health state [9]. In addition, [7] argue that pervasively observed and monitored individuals will find themselves pressured to alter behaviors in line with particular conceptions of health. The inhibition threshold is significantly lower when health is part of the decision-making. Solutions for a faster or better healing process are often considered necessities among patients or people with chronic diseases. Solutions connected to sustainable IoT are frequently implemented with fewer concerns as soon as they propose a faster healing process.

SHS increasingly support passive interaction mechanisms not visible to the naked eye, where users are led and supported during everyday tasks. This, in combination with previously active interactions where user input leads to output, results in a more intuitive living experience [44].

This chapter also addresses the interaction between security and privacy in IoT. Both are very significant regarding smart homes. The importance of robust encryption and strict safety protocols are highlighted, and there is a need for ongoing education about privacy and security risks. Additionally, adherence to data protection regulations like GDPR [42] enhances industry standards. At the same time, incorporating Privacy-Enhancing Technologies (PET) pushes toward a balanced approach for safer smart home environments [28].

Potential improvements to deal with the rapid growth of smart devices are described, including low-power IoT devices on 5G networks [12] and the advancements in Artificial Intelligence (AI) [3]. Future research is required to enhance low-power IoT solutions that can provide secure encryption methods and increasingly include AI's role, which presents another vast field of opportunities in fully yielding the potential offered by these new technologies.

This chapter is structured in the following way. A quick background overview is provided in section 6.3, where the IoT and the current state of technology are described. Section 6.4 describes various smart home features and protocols in place to guarantee successful and secure communication between devices, emphasizing the role of these protocols in addressing privacy concerns. Furthermore, SHS's environmental and health impacts are discussed in sections 6.4.7 and 6.4.6, respectively. Finally, an analysis of SHS is provided in section 6.5 before limitations and future work are presented in sections 6.6 and 6.7.

6.2 Motivation

The modern way of life has been transformed by the introduction of IoT technology, especially in terms of smart homes [38]. When such technologies knit together a complex network of devices, they bring unprecedented convenience and efficiency [47]. However, this accelerated integration also raises substantial concerns about personal data security and individual privacy [22]. These concerns primarily include the risk of unauthorized access to personal data and the potential for data breaches. The nature of IoT-enabled improvements, marked by both potential benefits and risks, requires carefully examining their effects on privacy. Despite its wide adoption, public concern is raised about protecting personal information, specifically in the smart home surroundings. Many users are worried about how their data is being used, who has access to it, and the lack of transparency from companies regarding data handling practices. This concern is amplified by frequent reports of data breaches and the misuse of personal information by third parties.

Many existing studies tend to focus on various aspects of smart homes [27, 8, 4]. This chapter captures it as a whole, presenting its extensiveness. Also, rapid technological development makes it hard to catch up, almost lagging behind, leaving room for poor evidence-based regulatory and technical responses. Therefore, it is essential to keep writing about and probing the subject.

Filling this gap requires a study that will not only comprehensively analyze the technological developments but also provide a critical review of the privacy issues surrounding these systems. Due to changes in privacy laws and new IoT devices being invented every other day, it becomes crucial to undertake a current and exhaustive review that would aid in assessing risks associated with these devices and recommending solutions that can be implemented.

This chapter aims to contribute to the existing knowledge on IoT and privacy by providing an in-depth understanding of how diverse IoT networks within smart homes affect privacy. It looks at different subsystems, such as smart lock systems and health monitoring devices, to comprehend their individual and cumulative effects on user privacy. Additionally, the chapter will evaluate the viability of prevailing legal frameworks like GDPR [42] meant to reduce privacy risks, which may provide helpful suggestions for future amendments or even formulation of new policies. Lastly, various aspects are discussed, including which future work could lead to considerable advancements in SHS.

6.3 Background

The following section provides a short overview and introduction to the IoT, pointing out various domains in which it is ingrained. Furthermore, the current state of known technology is described before delving into specific SHS features and their impacts.

6.3.1 Internet of Things (IoT)

Nowadays, a world without IoT devices is close to unimaginable. Figure 6.1 shows how big of an impact the IoT has and how many different domains it is involved in. In every aspect of daily life, IoT has its presence. It is helpful to highlight that it is divided into two sectors: the industrial and private sectors. Control and monitoring of machines, devices like traffic lights and rail switches, as well as replacements of treadmill work and support in factories plus healthcare, is considered the main impact on the industrial side. Referring to the consumer side, the management of smart devices such as phones, watches, and TVs, as well as smart homes, including sensors for tasks like energy regulation, presents the core of the IoT [28]. The latter is the core of this chapter.

6.3.2 Known Technology & Current Status

Smart homes can be understood as a connected and combined system of the many smart appliances that have communication access locally or via the internet [20]. Meanwhile, most of the devices used in a home can be bought as smart devices with additional integrated context awareness, communication capabilities, and autonomous computing [39]. This spans from larger systems such as air-conditioning & heating, audio & video streaming, and security all the way to simple devices like TVs, kettles, and toothbrushes.

Such a smart home aims to provide the best possible comfort to the end-user. This can be achieved in many ways by offering options such as preheating the oven over the phone while riding the bus home, lowering the blinds whenever the room temperature increases due to incoming sunlight, or adjusting the brightness of the lights depending on the time of day - all while optimizing energy consumption [28]. However, with these conveniences comes the need to address privacy concerns, as integrating these smart features involves collecting and processing personal data. It is crucial to ensure that data, such as user

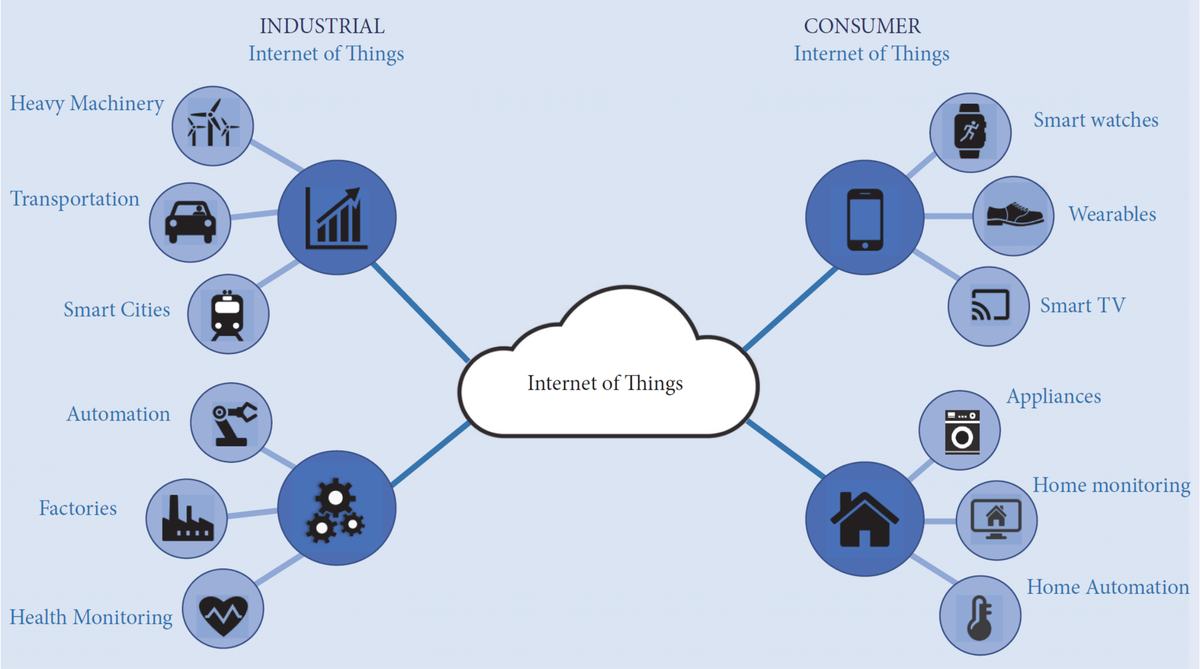


Figure 6.1: This figure presents various domains of the IoT [28].

habits, preferences, and schedules, is managed securely. Implementing robust privacy measures alongside these smart functionalities can help protect user information from unauthorized access and misuse, thereby maintaining the trust and security essential for widespread adoption.

However, since different factories with different hardware parts manufacture all these devices, it follows that not only are there high variations in IoT devices but also in the data collected, generated, and communicated [35]. In addition, these devices use various available protocols (such as Ultra-Wideband [1, 48], Bluetooth Low Energy [41, 37], or ZigBee [49, 13]), which are all implemented in different ways. This heterogeneity of IoT offers consumers a wide selection of choices and numerous possibilities for customization. However, it also introduces several challenges and open issues, such as interoperability, data security, and privacy concerns when using these systems [22, 35, 14, 4]. One significant challenge is ensuring user data privacy across this diverse ecosystem. With the varying standards and protocols, consistent and robust privacy measures must be implemented to protect sensitive information from unauthorized access and misuse. This diversity necessitates a comprehensive approach to security and privacy to maintain consumers' trust while leveraging the benefits of IoT technology.

6.4 Smart Home Technologies & Their Impacts

The following sections discuss various aspects of Smart Home Systems (SHS). Namely, their lock systems, Ultra-Wideband technology, environmental impact, impact on consumer health, and access management methods are highlighted. These topics were chosen due to their relevance and current priorities in enhancing security, efficiency, sustainability, and overall well-being in smart home environments.

6.4.1 Lock Systems

Smart devices designed to prevent unauthorized access into a physical space or as a digital barrier to sensitive data are increasingly in demand. Their growing acceptance is due to their ease of use and lower prices [2]. On the other hand, while these systems are attractive due to their affordability, the low cost often means that they have compromised hardware and software quality, ironically raising the risk of breaches despite being primarily meant to enhance the security [36]. As more unknown brands with more robust pricing strategies enter the market for these smart security solutions, concerns arise about the resilience of these security systems to well-organized attacks by more established firms. One key reliability aspect concerning such smart lock systems is compliance with recognized security standards. For example, the BS3621 standard requires a device to undergo a clear set of pre-defined tests before it can be rated secure. In contrast, the TS621 standard is not so strict since it only requires the product in question to outperform its testers. This disparity between standards highlights the importance of consumer awareness about safety specifications and certifications for devices used at home. Consumers must understand that standards with ”similar sounding names” can have fundamentally different expectations regarding requirements for the product in question [2]. Indeed, this rapid transformation of smart security devices necessitates better-informed customers. These tools need clear-cut guidelines and easily accessible information concerning their capabilities and weaknesses. The items from relatively obscure manufacturers that employ RFID and Bluetooth technologies are the most problematic. They could be breached through relatively simple hacking tools like Baofeng UV 5R radio or HackRF RFID NFC Card Copier [2].

Measures should be taken to mitigate these risks. There is a need for an increase in industry-wide promotion of improved security standards along with regular updates that would accommodate new technological threats. Regulatory bodies should ensure that all market actors adhere to these rules, thereby creating a level playing field irrespective of the brand. Knowledge gaps must be bridged through various educational initiatives. Consequently, this can be achieved by enlightening consumers about why security features matter, how they work, and which certifications are relevant when buying smart lock systems. There is a need to continue researching and developing more secure and resilient technologies. Innovations that enhance encryption methods, improve authentication processes, and integrate anomaly detection systems can significantly strengthen the security of smart lock systems. The manufacturers should regularly publish their security audits and updates as part of their commitment to transparency. Hence, simple yet straightforward communications would not only build trust but also enable customers to make informed choices regarding their security solutions. Indeed, smart lock systems have enormous potential for home safety. Nonetheless, navigating intricacies as well as vulnerabilities inherent in these systems is essential in ensuring that they serve their intended purpose without jeopardizing user safety or compromising privacy. By enhancing standards, educating consumers, and fostering innovation, the industry can safeguard against the risks while maximizing the benefits of smart security technologies.

6.4.2 Central-Access Management

Managing all the different devices in a smart home can be overwhelming, often leading to forgotten devices, which pose a security risk. Therefore, having a central management system for all IoT devices in a smart home is crucial. As IoT devices become more prevalent in homes, managing increasingly complex and diverse technologies becomes critical. Traditional mechanisms, such as remote controls attached to a wall (e.g., Philips Hue remotes), are no longer necessary since more advanced and integrated alternatives have taken over. It is now common practice to find a central control system for smart home devices, which comes from software applications installed on smartphones or smartwatches. These have eased the lives of people who can control their multiple appliances through Wi-Fi, whether within their buildings or away [44].

Ultra-Wideband (UWB) technology is one of the significant technological breakthroughs in this field, and its implications will be discussed more closely in the next section. Initially designed for military purposes, UWB has found extensive applications in domestic environments, particularly in smart homes. Secure Fine Ranging differentiates it from other technologies by allowing high accuracy tracing and elaborated security features such as Presence Detection, Follow-Me, and Point and Trigger Control, which are discussed by [44]. This feature resolves many practical problems SHS face, including how interactions between devices should be facilitated seamlessly. Additionally, UWB integration into conventional SHS has resulted in more robust and dependable house management ecosystems. Wireless routers at home facilitate interaction between remote control tools such as application servers, client computers, tablets, and iPads, making the IoT environment monitored, controlled, and managed 24 hours per day [48].

6.4.3 Ultra-Wideband (UWB) Technology

The Ultra-Wideband (UWB) technology has been fundamentally overhauled from its initial purpose of military communications. UWB was licensed for commercial use by the Federal Communications Commission (FCC) in 2002 and has since become a household name in daily technological lives [48]. It was initially meant to operate in the unlicensed range from 3.1 GHz to 10.6 GHz as an alternative to Wi-Fi. Still, it switched focus from data transfer, making secure ranging and localization its specialty due to power limits, which kept it from realizing its earlier promise of high bandwidth data transmission. The transition of UWB into a technology known for providing accurate, secure distance and location measurements came with significant improvements implemented in IEEE 802.15.4a and later IEEE 802.15.4z standards, respectively. Consequently, there has been a massive uptake of UWB in high-accuracy indoor Real-Time Locating Systems (RTLS), particularly after enhancing security provisions introduced via IEEE 802.15.4z [48].

Presently, UWB is distinguished not only by being another option for connectivity but also by focusing on secure fine-ranging capabilities that it optimizes uniquely, thus complementing existing solutions such as Wi-Fi, Bluetooth Low Energy (BLE) [41, 37], and

Near Field Communication (NFC); hence enabling easy integration with numerous other devices thereby expanding the technology's domain across multiple consumer and IoT platforms. Figure 6.2 shows how UWB is used on top of BLE to guarantee a successful access control mechanism, making use of the low-power signals used for authentication before enabling UWB for secure range processing [48].

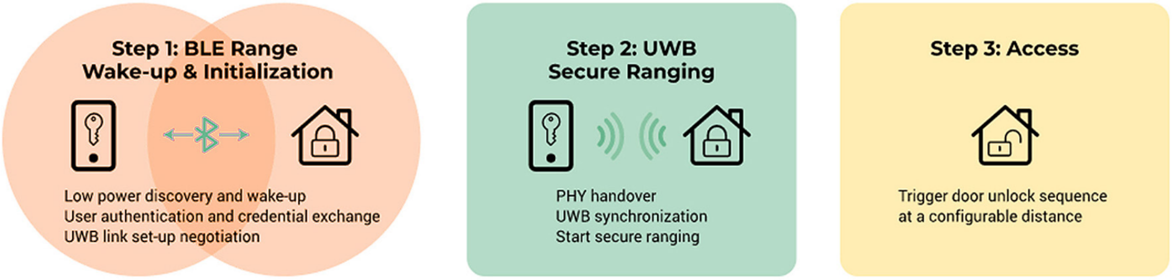

Step 1: BLE Range
Wake-up & Initialization
Low power discovery and wake-up
User authentication and credential exchange
UWB link set-up negotiation
Step 2: UWB
Secure Ranging
PHY handover
UWB synchronization
Start secure ranging
Step 3: Access
Trigger door unlock sequence
at a configurable distance

Figure 6.2: This figure taken from [48] shows how UWB manages to make use of BLE, which is already built into many smart devices to enable a secure access control mechanism.

This integration marked a significant milestone in developing UWB-supported devices, positioning UWB to break out as a mainstream solution for fine-ranging precision positioning. With its ability to provide centimeter-level distance and location measurement accuracy, UWB is set to revolutionize several secure ranging and positioning applications, making it a vital building block for tomorrow's IoT connectivity [48]. Moreover, the enhanced precision and security of UWB technology contribute to better privacy protection by ensuring that only authorized devices and users can access sensitive location data, thereby mitigating potential privacy risks in smart home environments.

6.4.4 Multi-Access Edge Computing (MEC)

MEC is the approach to managing data traffic and processing loads of data from IoT devices in smart homes. To address the challenge in terms of latency, bandwidth use, and privacy associated with traditional cloud-centric models, MEC locates its data processing closer to where it comes from at the edge of a network [46]. By processing data locally, MEC enhances privacy by reducing the need to transmit sensitive information over long distances, thus minimizing exposure to potential breaches. Latency and bandwidth remain a considerable concern when referring to smart homes that require real-time decision-making and data processing for them to be responsive or automated. Sometimes, these traditional cloud-computing models may take longer to process this information, resulting in unacceptable delays and high bandwidth consumption. Network congestion leads to slowed response times with increasingly interconnected IoT devices [31]. In such situations, MEC processes the data at or near its source of origin, which can occur within localized data centers or at radio tower stations [23]. Such proximity requires shorter distances for data transmission, thus reducing latency and leading to faster decision-making, which is especially crucial in safety systems, energy management, and emergency responses within smart homes. Additionally, this local processing enhances privacy by minimizing the exposure of sensitive data to potential breaches during transmission, ensuring that personal information remains secure and compliant with privacy regulations. Another side effect of MEC is enhanced privacy and security, bridging the gap to smart homes. By keeping all private information local, MEC reduces exposure to breaches and cyberattacks. Storing data locally allows personal information to be stored more securely by conforming strictly to regional laws like GDPR [42] or nFADP [40]. This impacts privacy by significantly reducing the risk of unauthorized access and data breaches, as sensitive information remains within a controlled and regulated environment. Additionally, local data storage ensures that personal data is handled according to specific legal and ethical standards, thereby enhancing trust and compliance with privacy expectations. This approach not only protects individual privacy but also strengthens the overall security framework of smart home systems.

Overall, a reduction in network load can be achieved through implementing MEC. Keeping data processing within the localities where it is created will require only a few communications to be sent back to central data centers or clouds. This reduction in data traffic over the network increases the bandwidth for other uses and generally improves network resource utilization. Furthermore, by processing and storing data locally, MEC enhances privacy by minimizing the exposure of sensitive information to potential breaches during transmission and ensuring compliance with regional data protection regulations, such as GDPR in the European Union [26].

MEC can reduce complexity and costs related to cross-border data transfers by ensuring that data is processed and stored within the region it was generated. This regional approach not only enhances data privacy but also ensures that host countries' data protection regulations are strictly followed. A distributed computing architecture is developed by adopting MEC for smart homes, whereby multiple small computing centers process the data. This makes networks more reliable because they do not depend on single nodes, and scalability becomes possible, allowing new nodes to be easily incorporated as IoT devices increase in number [23]. By keeping data local, MEC supports a robust privacy framework, protecting user information and enhancing the overall security of smart home systems.

In conclusion, Multi-Access Edge Computing (MEC) enhances the capabilities of smart home IoT networks by improving data privacy and security. By processing data locally, MEC reduces the need to transmit sensitive information over long distances, thereby minimizing exposure to potential breaches and cyberattacks. This local processing also ensures compliance with local data protection regulations, such as GDPR, further safeguarding user privacy. Additionally, MEC reduces latency and bandwidth usage, leading to more responsive and efficient smart home systems. By creating a highly efficient, secure, and robust IoT ecosystem, MEC represents an essential milestone towards sustainable smart homes that can meet emerging needs while maintaining high standards of privacy and security [31].

6.4.5 128-Bit AES-CCM Algorithm

When smart home technologies increasingly become part of daily lives, there is a need to transmit data within these systems. The 128-bit AES-CCM (Advanced Encryption Standard - Counter with Cipher Block Chaining Message Authentication Code) algorithm is a robust cryptographic protocol for encryption and authentication. This gives it an advantage over many other protocols when securing network communication for IoT devices, such as those in smart homes [10]. This is a symmetric block cipher that the U.S. government adopted as an encryption standard. It has been widely used worldwide because of its hardware and software efficiency, as well as high security. This symmetrical key algorithm uses one key for both the encrypting and decrypting processes. It functions on fixed block sizes of 128 bits with keys being 128, 192, or even 256 bits long [5]. CCM stands for ”Counter with Cipher Block Chaining Message Authentication Code”. This mode combines counter mode (CTR) with CBC-MAC for authentication purposes. While mixing these two methods might acknowledge the scrambled message, this encryption mechanism mainly ensures the safety criteria. The initial block configuration sets up a nonce (a number that can be used only once per session) and a counter to start things off for AES-CCM. For instance, in the plain text data, the CTR method depends on this configuration [10]. If bytes are arranged on the grid, AES uses another table to substitute each byte in the grid with another one. The cycles of these rows shift according to different number counts per row, giving more complexity. Diffusion across columns is enhanced by mixing each column of bytes using an invertible linear transformation. The block receives a subkey derived from the primary key using the XOR operation. This step continues through several rounds: ten for 128-bit keys, 12 for 192-bit keys, and 14 for 256-bit keys [5]. Post-encryption CBC-MAC mode helps generate a message authentication code (MAC) that can be used to check integrity and authenticity when decrypting the message. Using AES-CCM in smart home devices accommodates the need for privacy and reliability. Sensitive information, such as details on security in homes or personal data, should remain confidential by encrypting information sent between devices by AES-CCM. Also, AES-CCM checks that messages have not been changed, thus maintaining data integrity [10]. Understanding the detailed encryption and authentication processes of AES-CCM is essential because it underscores the algorithm’s role in safeguarding privacy in smart home systems. By ensuring that data remains secure and unaltered, AES-CCM supports a reliable and private smart home environment, addressing privacy concerns and enhancing overall system security.

To sum up, the 128-bit AES-CCM algorithm is a perfect solution for securing IoT devices in smart homes by balancing solid security requirements with the limitations posed by IoT devices. Its deployment serves a dual purpose: protecting against potential cyber threats and forming an integral part of overall smart home security strategies. Moreover, AES-CCM enhances privacy by ensuring that sensitive data remains encrypted and authenticated, preventing unauthorized access and tampering. This robust encryption mechanism safeguards personal information, creating a secure and private smart home environment [33].

6.4.6 Environmental Impact & Sustainability

Achieving sustainability marks a significant milestone in the evolution of environmentally friendly IoT smart home devices, demonstrating how technological advancements can reduce energy use and increase efficiency. Integrating these technologies into homes, cities, and vehicles reshapes the energy landscape toward a more sustainable future. Therefore, the impact of smart technologies extends beyond individual homes, contributing to broader environmental initiatives. Consequently, smart cities, which include smart homes, integrate similar IoT technologies to manage everything from traffic flows to public utilities, thereby reducing carbon footprints and improving the efficiency of urban infrastructures. This holistic approach makes the entire living ecosystem more efficient and optimized regarding energy consumption, ultimately maximizing the use of natural resources [19].

However, as smart homes and cities become increasingly interconnected, the importance of data privacy cannot be overstated. The extensive use of IoT devices results in collecting and processing vast amounts of personal data. Protecting this data is crucial to maintaining user trust and safeguarding privacy. As we build more integrated and sustainable environments, robust privacy measures must be implemented to protect the sensitive information collected and used. This includes ensuring that data is encrypted, access is controlled, and compliance with data protection regulations is maintained. Figure 6.3 shows how smart homes are a fundamental building block for future smart cities. Therefore, interoperability, flexibility, decentralization, and robust privacy protections are of utmost importance to ensure both the security and privacy of the data collected within this interconnected ecosystem. By addressing these privacy concerns, we can ensure that the benefits of smart technologies are fully realized while maintaining the trust and confidence of the users.

In the residential sector, the eco-friendly transition heavily relies on technological advancements such as smart homes. These households have sophisticated energy management systems that monitor real-time consumption patterns while adjusting usage based on forecast models or other data sets. The result is optimal energy usage without compromising comfort or convenience. For instance, automated thermostats, low-energy lighting systems, and power-saving appliances collectively reduce overall power requirements within households.

The fundamental components needed for this transformation are innovative technological applications resident at the heart of a smart home, like real-time monitoring of energy consumption using diagnostic tools such as sensors, which can offer detailed insights about homeowners' electricity behaviors so that they can utilize their electricity judiciously. However, these technologies also collect a vast amount of data, raising essential privacy concerns. Ensuring this data is protected from unauthorized access is crucial to maintaining user trust.

Moreover, integrating renewable sources like solar panels linked with geothermal systems will facilitate homes generating their green power while significantly decreasing dependence on fossil fuels [19]. As these systems become more advanced, robust privacy safeguards must be implemented to protect the sensitive data generated by these smart

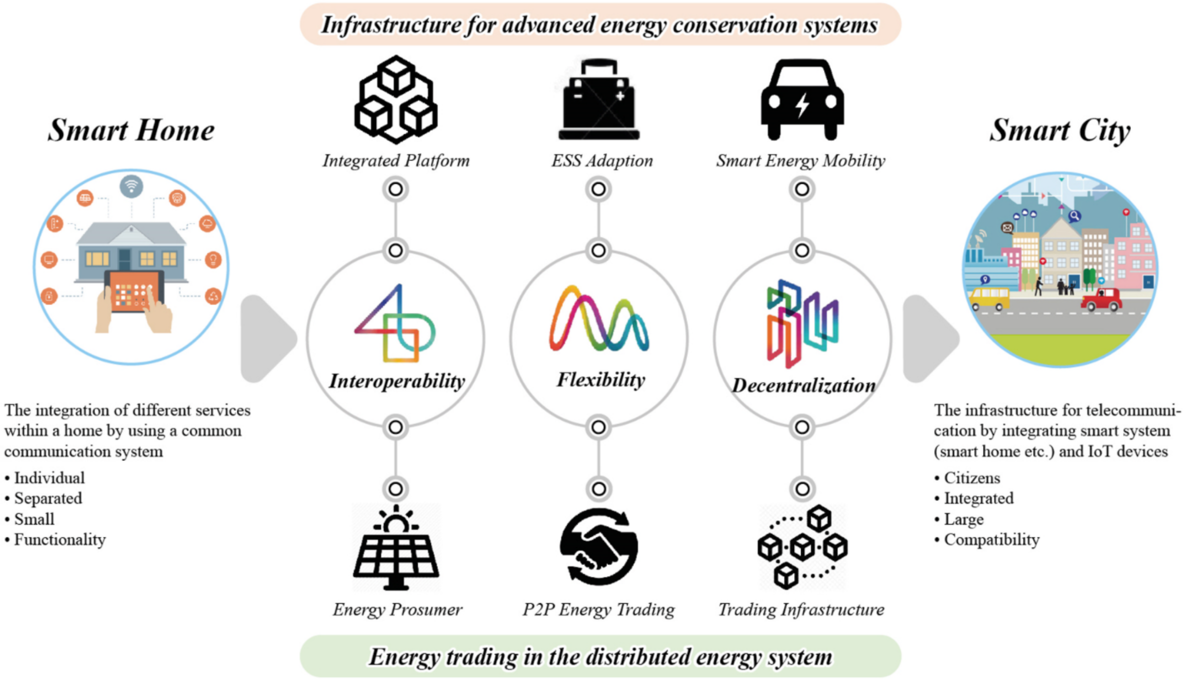

Infrastructure for advanced energy conservation systems
Smart Home
Integrated Platform
ESS Adaption
Smart Energy Mobility
Smart City
Interoperability
Flexibility
Decentralization
The integration of different services within a home by using a common communication system
• Individual
• Separated
• Small
• Functionality
The infrastructure for telecommunication by integrating smart system (smart home etc.) and IoT devices
• Citizens
• Integrated
• Large
• Compatibility
Energy Prosumer
P2P Energy Trading
Trading Infrastructure
Energy trading in the distributed energy system

Figure 6.3: This figure taken from [19] shows the importance of integrating sustainable technologies into smart homes, eventually leading to sustainable smart cities.

technologies. Balancing technological innovation with privacy protection is essential to ensure eco-friendly smart homes' widespread adoption and success.

However, this perspective is changing because they are becoming part of the solution rather than just being energy consumers as it has been traditionally thought. Smart homes have a dual role through energy optimization and waste reduction; they minimize their environmental footprint and supply excess power from renewables to the grid. Thus, the next task will be to improve the scalability of smart home technologies and make them compatible with larger smart grids. Consequently, future trends should ensure increased compatibility amongst devices and systems to achieve more significant energy savings and sustainability gains. Additionally, data privacy and security will remain of prime concern as technology evolves in an increasingly interconnected world [15].

IoT plays an increasingly central role in pushing forward environmental sustainability through the use of smart home technologies. Therefore, during this advancement period, it is imperative to stress open standards and sustainability in these technologies to make them effective contributors to global efforts against climate change. By adopting IoT for more intelligent energy management, residential spaces from simple users are revolutionized into active actors fighting for a sustainable planet by adopting IoT for more intelligent energy management [19, 15].

The environmental well-being of smart homes, smart cities, and the planet significantly contributes to every individual's health. The following section will further explore this crucial connection.

6.4.7 Health Impacts

By 2040, the rate of older people over 60 will be around 21% globally [25], enhancing the need for IoT devices that ensure their safety and independence. Nonetheless, these technologies are not only beneficial to the aged but also helpful for those dealing with temporary illnesses, chronic diseases, and disabilities. These smart devices are crucial for helping these groups live independently, assisting them with complex tasks, and providing essential reminders, ultimately enhancing their quality of life [25].

With improvements in information and communication technology, smart home technologies substantially improve health care within supportive living environments by integrating into users' daily routines. Devices that sense and monitor their environment have rapidly evolved with various social needs. For example, Z-Wave and ZigBee [49, 13] technologies enable smart homes to automatically adjust their functions based on usage patterns, thereby optimizing energy consumption and minimizing utility costs [25]. Additionally, such devices can alert healthcare facilities during emergencies, contributing to enhanced safety and response time.

Health monitoring technologies involve collecting and analyzing data from human body conditions such as blood pressure, glucose levels, respiration rates, body temperature, and other vital signs [16]. Wristbands or even smartwatches can keep track of everyday activities, making them ideal for monitoring emotional states and providing fall prevention mechanisms, like emergency detection capabilities [11]. Safety monitoring devices are designed to detect environmental hazards, including gas leakages or fires, which trigger emergency protocols to guarantee residents' safety. The overall security of the living environment is improved by security monitoring systems that help identify potential human threats or suspicious activities [25].

However, the integration of such technologies raises significant privacy concerns. As these devices continuously collect and transmit sensitive health and personal data, ensuring that this information is adequately protected against unauthorized access and breaches is crucial. Ensuring compliance with data protection regulations, such as GDPR, can help safeguard user privacy.

Technology has also supported increased social interaction, allowing virtual engagements with friends and family. For instance, seniors can make video calls to stay connected with grandchildren and other family members, thus enhancing feelings of inclusion, especially during difficult times like the recent pandemic. Cognitive and sensory assistance technology enabled by machine learning provides practical assistance in daily activities. The technology can remind users about their medication timetable or guide them to where they want to go within and outside the house [34]. While these features offer significant benefits, they also require robust privacy measures to ensure the data collected is used responsibly and securely.

The integration of IoT in healthcare holds the promise of more innovative developments that could redefine traditional care models. How health monitoring will be done in smart homes is expected to change significantly over time as it becomes more proactive. Developing advanced technologies that predict health events before they materialize through analyzing health trends using sophisticated algorithms can provide timely warnings to both users and healthcare providers. This proactive approach could significantly shift how health is managed at home, offering a more predictive, personalized healthcare experience that keeps individuals safe, healthy, and well-informed about their health [17]. Ensuring the privacy and security of the data involved in these processes is paramount to maintaining user trust and protecting sensitive health information.

6.5 Analysis of Smart Home Systems

The following section discusses the two modes of interaction with SHS. In addition, the security of keeping sensitive data protected is compared to privacy when collecting relevant data. Lastly, various possible improvements for the discussed topics are presented.

6.5.1 Active vs. Passive Interaction

The IoT has paved the way for smart homes. Previously, such dwellings were fitted with intelligent control systems that require little or no human intervention. These devices have enhanced the interaction between users and their living spaces by evolving from passive to more active and ever-present input methods in these places. This transition is not only what makes them better but also enables a mix of active and passive interactions, making room for different preferences [34]. Usually, homeowners give direct instructions or perform specific actions to manipulate gadgets in the house. People can command using voice commands, touchscreen interfaces, and keyboards, enabling dynamic interaction with the system. Users can check their schedule, ask about weather conditions, or instruct a home theater system on what content they want to play. Pre-defined processes or routine reactions occur when certain environmental factors are met without direct intervention. Motion sensors could turn on lights automatically if someone enters a room. At the same time, smart thermostats can adjust temperatures depending on the time of day or presence of people, even more accessible today with an omnipresent fine-range Ultra-Wideband Net on the horizon [48]. Passive systems function in the background mode for improved comfort levels that enhance convenience and optimize energy usage without disturbing user routines.

Joining active and passive systems inside a smart home ensures an intuitive living experience. Sometimes, according to previous choices made by occupants, lighting may be adjusted passively to comfortable levels, while some manual commands given via voice recognition or apps can still change it. In return, this results in higher level accommodation with personalization, allowing it to adapt over time based on the habits and preferences of the residents [44].

In the foreseeable future, smart homes will have more passive components since they will be driven by artificial intelligence that is far more advanced. Such systems would anticipate user needs and make adjustments even before they are explicitly commanded, for example, preheating an oven at a particular time or adjusting lighting according to sunlight and user schedule. Additionally, machine learning advancements combined with user identification technologies can help smart homes differentiate between users, leading to personalized environments that self-adjust according to each person's preferences without requiring them to actively participate in setting up their conditions [18, 3]. Figure 6.4 describes an AI framework that can eventually provide a personalized system.

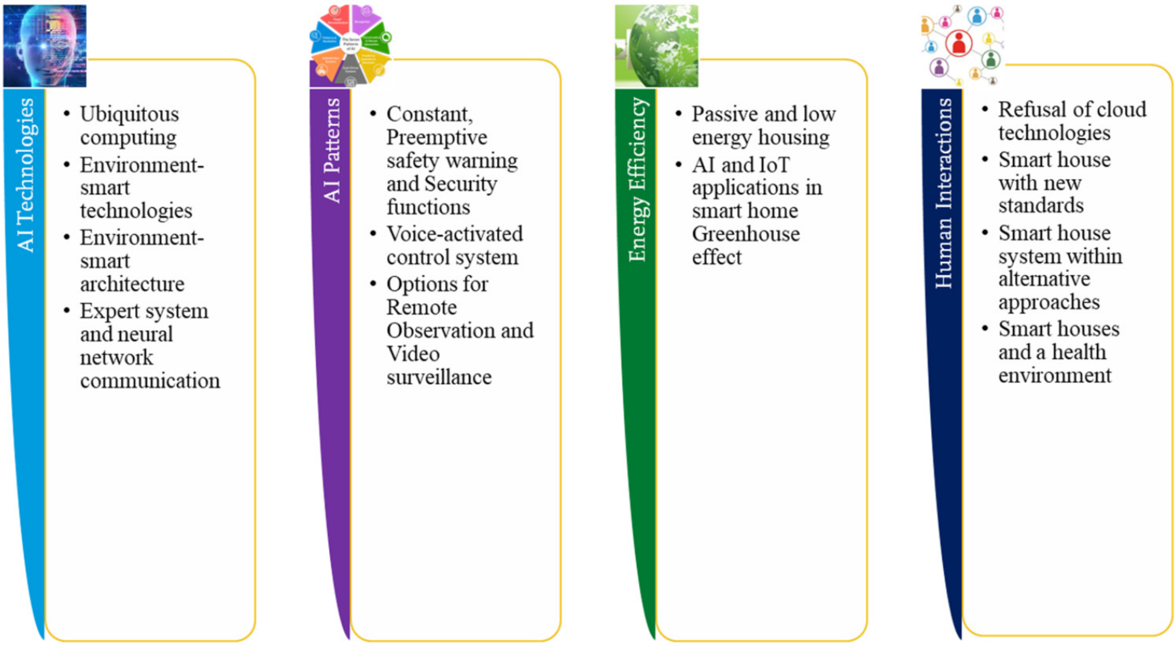

AI Technologies
• Ubiquitous computing
• Environment-smart technologies
• Environment-smart architecture
• Expert system and neural network communication
AI Patterns
• Constant, Preemptive safety warning and Security functions
• Voice-activated control system
• Options for Remote Observation and Video surveillance
Energy Efficiency
• Passive and low energy housing
• AI and IoT applications in smart home Greenhouse effect
Human Interactions
• Refusal of cloud technologies
• Smart house with new standards
• Smart house system within alternative approaches
• Smart houses and a health environment

Figure 6.4: This figure taken from [3] shows a framework of AI application concerning smart homes.

As the number of passive devices in smart homes increases, they are often harder to observe and less present in daily minds, making it even more critical to ensure solid and reliable security within these systems. The increased invisibility of such devices heightens the need for robust privacy measures to protect sensitive data and prevent unauthorized access. Consequently, the next section will focus on Privacy and Security in Smart Homes, addressing the essential protocols and strategies required to safeguard users' functionality and personal information in an increasingly interconnected environment.

6.5.2 Security vs. Privacy Concerns

The interaction between security and privacy in IoT, especially regarding smart homes, is an elaborate space that needs continuous rethinking and a lot of consideration because the landscape of threats and vulnerabilities is constantly evolving. As more devices become interconnected and collect vast amounts of personal data, the potential for security breaches and privacy invasions increases. This section summarizes the existing security issues, explores possible solutions, and discusses ways to make these concepts more straightforward for end-users. By addressing both the technological and human factors, we can develop more resilient systems that protect user data while maintaining the convenience and efficiency that smart homes promise.

The first priority should be security in SHS due to the risk of hacking and intrusion on devices or collecting data. Many IoT devices still have inadequate security measures, which makes them vulnerable to cyber-attacks. These vulnerabilities harm people's privacy by exposing their private data and compromising general safety when things like smart locks are breached [12, 14]. Therefore, robust encryption methods must be developed alongside strict safety protocols. All manufacturers must ensure that their devices follow the most stringent security requirements from design up to end-of-life of such products. Regular firmware updates should be made so that new threats can always be addressed. Managing these risks requires educating consumers about privacy and security issues related to IoT devices. Users need access to educational materials free from complex jargon explaining how others collect, store, or use their information. Workshops, online tutorials, or interactive guides may assist, making it easy for ordinary people to comprehend. Also, public awareness campaigns and manufacturers' transparency reports can build consumer trust. It must be essential to show, in general, how much effort can be put into safeguarding user privacy and protecting it from unwanted leaks [14].

Regulations such as the General Data Protection Regulation (GDPR) [42] and the new Federal Act on Data Protection (nFADP) [40] play a significant role in shaping industry standards and practices, particularly in the realm of IoT and smart homes. These laws mandate strict data protection and privacy guidelines, holding companies accountable for security breaches and privacy violations. Adhering to these regulations not only safeguards consumers against harm but also fosters trustworthiness and reliability in IoT technologies [6].

In the context of smart homes, these regulations ensure that personal data collected by IoT devices, such as security cameras and smart thermostats, is handled securely and used responsibly, thereby protecting user privacy and enhancing consumer confidence in smart home technologies.

In the future, incorporating Privacy-Enhancing Technologies (PETs) like homomorphic encryption or zero-knowledge proofs in IoT devices promises to be an exciting area. This means such technologies allow data processing and analysis while keeping it confidential, thus solving the privacy-security dichotomy. Therefore, achieving a balance between security and privacy in smart home IoTs is always a matter of concern which requires multiple approaches. Combining technological solutions, consumer education, and robust regulatory frameworks can protect users while encouraging innovation and convenience in developing smart homes [28].

6.5.3 Potential for Improvement in Various Fields

The fast expansion of IoT technology has been chiefly driven by its incorporation into consumer items that automatically collect and relay data. In addition to making personal IoT devices more readily available, this growth has also empowered people to know more about their health, become more efficient at what they do daily, and automate their everyday operations, thereby improving life's quality. Figure 6.5 shows how the number of connected IoT devices has drastically increased over time.

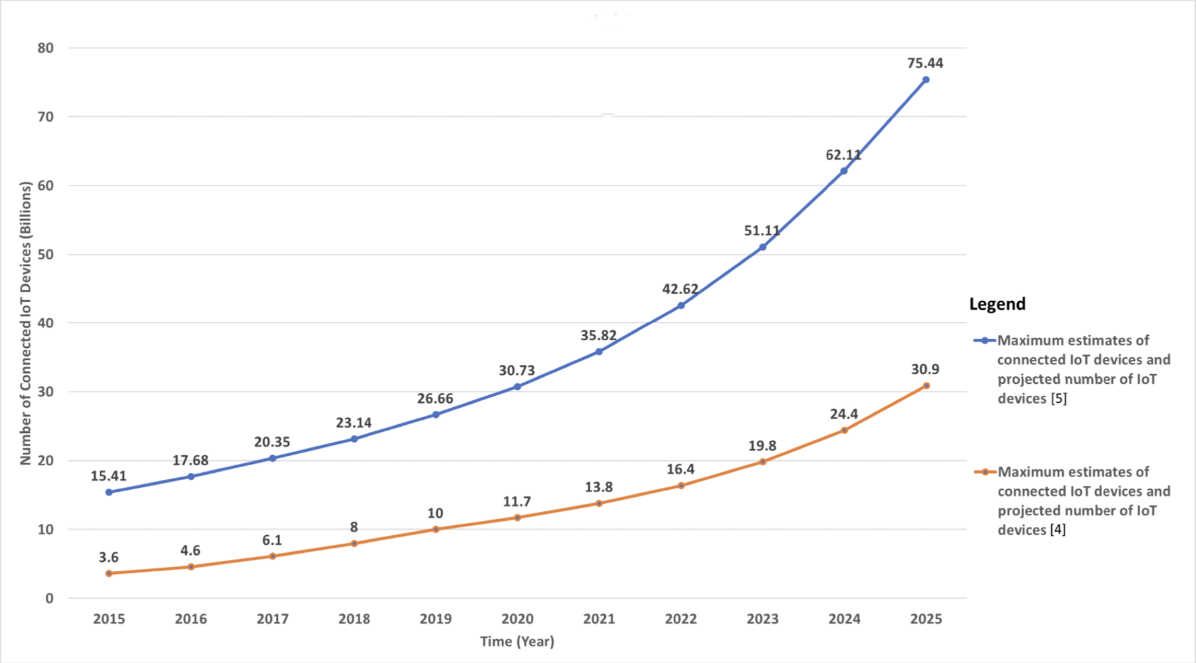


Figure 6.5: This figure shows how the number of connected IoT devices has grown over the past 20 years [12].

While the adoption of IoT steadily rises, several areas require further improvement, particularly in terms of security and privacy. For instance, low-power IoT devices on 5G networks represent a significant opportunity for enhancement. These networks can improve device functionality and efficiency through faster data speeds and more reliable connections [12]. However, they also raise new data security and privacy issues due to their extensive coverage and numerous access points.

Integrating advanced language models, such as hypothetical newer GPT versions, into smart homes presents considerable potential. These models could significantly enhance natural language understanding between humans and devices, allowing for more intuitive interactions. Additionally, their ability to handle massive quantities of data could play a crucial role in predictive maintenance and personalized user experiences. However, this also introduces privacy concerns, as the vast amount of data processed must be securely managed to protect user information.

Thanks to technological advancements, hardware components are continually reducing in size, enabling the realization of smaller, smarter homes with concealed features. Advancing nanotechnology innovations and materials is critical, as they may lead to smaller yet energy-efficient devices without compromising performance. This development highlights the need for stringent privacy measures to ensure that the data collected and processed by these advanced technologies is protected.

Another significant area of progress is Ultra-Wideband (UWB) technology, which has made substantial leaps in accuracy tracking and sensing surrounding areas for IoT devices. Emerging wireless communication technologies, such as Light Fidelity (Li-Fi) [30] and the new 5G NR (New Radio) [43] positioning techniques, also show promise by offering greater accuracy, lower power consumption, and improved security features. These advancements necessitate robust privacy safeguards to protect the vast amounts of data generated and transmitted.

Several future directions are available to explore as the boundaries of what is possible with IoT continue to be pushed. These include developing alternative encryption methods and data privacy safeguards to meet changing needs, establishing uniform standards to guarantee compatibility and seamless interaction between various IoT devices and ecosystems, and encouraging greener, more sustainable IoT solutions that reduce environmental impact while maximizing efficiency. IoT's full potential can be realized by addressing these aspects transforming everyday activities while ensuring user privacy and sustainability.

6.6 Limitations

Future computing, especially in the field of IoT and smart home devices, will require significant energy management and security innovations. These limitations have made it challenging to develop these technologies further and increase their adoption on a broader scale. Running multiple IoT devices continuously in a smart home is still a significant obstacle, regardless of advances in IoT technology [20]. This concern is particularly acute in the context of global environmental issues and the increasing need for sustainability. The machines must either migrate into working efficiently with renewable energy or reduce their power consumption levels immensely [19].

As smart home devices become more integrated into daily lives, there are increased chances for privacy breaches and security vulnerabilities. The challenge lies in robust security protocols adapting to changing threats while safeguarding user privacy [32, 22]. Additionally, increased integration between IoT devices with advanced language models could eventually provide new vector points for potential attacks; therefore, it is necessary to have higher safety standards coupled with constant care [3].

A dominant limitation within the current ecosystem around smart homes is the lack of interoperability between different manufacturers' devices [29]. The fragmentation limits seamless functioning within smart homes while complicating user experience. Overcoming these challenges means developing universal communication and compatibility standards, enabling an interconnected environment to emerge.

It is challenging for regulations to keep up with the rapid advancements made by IoT. This lag creates a complex landscape for manufacturers and consumers, who must navigate an often unclear regulatory environment. In many jurisdictions, ethical dilemmas such as data ownership, the right to privacy, and informed consent are not adequately addressed, making it harder for people to adopt or embrace this smart home concept. Even if it got better with the current updates of nFADP [40] and GDPR [42], keeping up with the current evolution is constantly challenging.

The existing technology constraints also hinder the current IoT devices in terms of processing power, battery life, and data storage. Their pace must be accelerated as these technologies evolve to meet the future demand for more complex applications. Technological breakthroughs will be essential in addressing these technological barriers through the development of components that are more powerful, efficient, and of a smaller scale [35].

Lastly, there is still a challenge regarding consumer acceptance and awareness of smart homes. Many potential users remain skeptical of smart home technology, often due to complexity, reliability, and cost concerns [36]. Widening adoption involves challenging such perceptions through improved marketing practices like transparent communications and showing how IoT integration brings real benefits.

In conclusion, although the future looks bright for smart home technology, its maximum potential can only be realized if these limitations are overcome. The future evolution of IoT devices depends not only on technological innovation but also on how their ethical, environmental, and security aspects are handled.

6.7 Future Work

Many future research and development possibilities are available because of the introduction of IoT technologies into smart homes. The current efforts have put a strong foundation. However, many areas still need to be explored further for these systems to improve their functionality, security, and user acceptance.

Further research on low-power IoT devices is crucial. Since the number of devices in an average smart home has increased, energy efficiency is becoming more critical than ever before. Energy harvesting techniques, which transform environmental energy (for example, solar, thermal, or kinetic energy) into usable electrical power, could significantly prolong the battery life of IoT devices. Besides, combining smart homes with renewable energy sources not only conforms to global sustainability goals but also reduces the overall energy footprint of these technologies.

Regarding security, developing new encryption methods and secure data storage solutions is vital to protect users' data from possible breaches [22]. Because IoT devices typically collect sensitive personal information, it should be a top priority to ensure the security of this information. Preserving privacy in technology, especially one that can be applied without interfering with the existing tools, will play a significant role in maintaining user trust and complying with regulatory standards.

The role played by AI in smart homes is expected to increase, which may result in more customized, adaptive, and intuitive surroundings in the future [3]. Research should seek better AI models that understand and predict human preferences and behavior to make interactions with smart home appliances more natural. This means improving natural language processing capacities behind voice-interface controls as well as devising more sophisticated predictive algorithms for home automation.

Besides technical aspects, prospective works must consider social, ethical, and psychological dimensions associated with living within fully connected automated environments.

This covers the effects of smart homes on social interactions, mental health, and daily routines. Ethical aspects such as data privacy, consent, and surveillance must be critically examined to ensure that smart home technology development conforms to societal norms and values. With technological advancement comes the need for advancing regulatory frameworks governing their use. Future research should engage policymakers in developing standards and regulations to ensure safe and reliable smart home technologies. These will include clear guidelines on data usage, device interoperability, and security standards.

The adoption of smart home technologies by many people will require joint efforts to educate consumers effectively. Future undertakings should aim at demystifying the technologies involved and explaining what is beneficial or risky about them by providing accessible, clear information that allows a consumer to make reasoned choices. Moreover, investigating market trends and consumer behaviors could provide insights into how best products can be customized to users' changing needs and expectations.

In conclusion, it may be observed that although numerous technical breakthroughs have occurred, allowing smart homes to enhance lives each day, fully utilizing these opportunities means tackling several technical challenges alongside complex social and ethical issues. The heterogeneity of IoT devices presents a unique challenge in ensuring compatibility and security across different manufacturers and platforms, directly impacting user privacy and data protection. Addressing these privacy concerns is critical as smart homes increasingly rely on interconnected devices that collect and process vast amounts of personal information.

Through cross-cutting research and collaborations among various industry actors, it is possible to develop robust solutions that address both the technical and privacy challenges posed by heterogeneous IoT environments. This collaborative approach can lead to the next generation of smart home technology that balances innovation, user-centricity, sustainability, and privacy protection. By prioritizing security and privacy, we can foster trust and confidence in smart home technologies, ensuring they continue to enhance lives while safeguarding personal data.

6.8 Conclusion

The summary of the existing literature has dealt exhaustively with the multifaceted relations between heterogeneous IoT networks and privacy in the smart home context. As IoT becomes part of daily lives, it brings improved convenience and efficiency but also significant challenges and risks, particularly regarding privacy and security.

From a vulnerability approach, it becomes clear that while IoT technologies significantly benefit automation and energy efficiency, they also present substantial risks by potentially exposing sensitive personal information. These gaps particularly concern less-known IoT device manufacturers and commonly used security devices, underscoring the need for tighter security approaches and strong privacy safeguards. This chapter also highlights that advanced regulatory frameworks, such as GDPR or nFADP, can help consumers by providing high standards for data protection, thereby addressing the privacy implications of heterogeneous IoT networks.

Further analysis indicates that consumers need education on how IoT technologies impact their privacy. Individuals should be aware of the dangers related to devices being hacked or infected with malware. Given the dynamic nature of IoT technology, continuous monitoring of developments, coupled with regular updates to security protocols, is indispensable to maintain privacy in an environment with diverse devices and manufacturers.

Future research must ensure more secure IoT architectures, especially those integrating privacy-enhancing technologies. New encryption methods, dependable access control systems, and intuitive interfaces for viewing and monitoring personal information will be essential. Lastly, within smart homes where IoT devices increasingly communicate with each other, research must explore the implications of these interactions on user privacy and data safety.

All in all, while there are undeniable advantages associated with expanding IoT devices in smart homes, a balanced approach is necessary to address the corresponding privacy and security issues. Therefore, the chapter seeks collective action among researchers, producers, policymakers, and end-users to achieve sophisticated levels of security and privacy within smart homes. By fostering transparency, continuously innovating, and respecting privacy, the opportunities presented by IoT within smart homes can be maximized while minimizing vulnerabilities associated with their heterogeneity.

Bibliography

[1] Mohamed Adeeb Ahmed. “Privacy Issues of Mobile Phone Companies’ Usage of Ultra-Wideband (UWB) Technology”. Master Thesis. Delft University of Technology, 2021. URL: http://resolver.tudelft.nl/uuid:6b147beb-918e-4ff5-a477-89fde89eb707.

[2] Ashley Allen et al. “Smart Homes Under Siege: Assessing the Robustness of Physical Security Against Wireless Network Attacks”. In: *Computers & Security* 139 (2024), p. 103687. DOI: 10.1016/j.cose.2023.103687.

[3] Amjad Almusaed, Ibrahim Yitmen, and Asaad Almssad. “Enhancing Smart Home Design with AI Models: A Case Study of Living Spaces Implementation Review”. In: *Energies* 16.6 (2023). DOI: 10.3390/en16062636.

[4] Z.A. Almusaylim and N. Zaman. “A Review on Smart Home Present State and Challenges: Linked to Context-Awareness Internet of Things (IoT)”. In: *Wireless Networks* 25 (2019), pp. 3193–3204. DOI: 10.1007/s11276-018-1712-5.

[5] J.J. Amador and R.W. Green. “Symmetric-Key Block Cipher for Image and Text Cryptography”. In: *International Journal of Imaging Systems and Technology* 15.3 (2005), pp. 178–188. DOI: 10.1002/ima.20050.

[6] G. Birchley et al. “Smart Homes, Private Homes? An Empirical Study of Technology Researchers’ Perceptions of Ethical Issues in Developing Smart-Home Health Technologies”. In: *BMC Med Ethics* (2017), pp. 18–23. DOI: 10.1186/s12910-017-0183-z.

[7] Ian Brown and Andrew A. Adams. “The Ethical Challenges of Ubiquitous Healthcare”. In: *International review of information ethics* 8 (2007), pp. 53–60. DOI: 0.29173/irie98.

[8] Joseph Bugeja, Andreas Jacobsson, and Paul Davidsson. “PRASH: A Framework for Privacy Risk Analysis of Smart Homes”. In: *Sensors* 21.19 (2021). DOI: 10.3390/s21196399.

[9] Jonathan Buonocore et al. “Health and climate benefits of different energy-efficiency and renewable energy choices”. In: *Nature Clim Change* 6 (2016), pp. 100–105. DOI: 10.1038/nclimate2771.

[10] Injun Choi, Jong-Yeol Lee, and Ji-Hoon Kim. “Design of Low-Complexity 128-Bit AES-CCM* IP for IEEE 802.15.4-Compatible WPAN Devices”. In: *Journal of IKEEE* 19.1 (2015), pp. 45–51. DOI: 10.7471/ikeee.2015.19.1.045.

[11] Matthew Clark et al. “Affordable Remote Health Monitoring System for the Elderly Using Smart Mobile Device”. In: *Sensors & Transducers* 184.1 (2015), pp. 77–83. URL: https://www.proquest.com/scholarly-journals/affordable-remote-health-monitoring-system/docview/1656619859/se-2.

[12] Jonathan Cook, Sabih Ur Rehman, and M. Arif Khan. "Security and Privacy for Low Power IoT Devices on 5G and Beyond Networks: Challenges and Future Directions". In: *IEEE Access* 11 (2023), pp. 39295–39317. DOI: 10.1109/ACCESS.2023.3268064.

[13] Parneet Dhillon and Harsh Sadawarti. "A Review Paper on Zigbee (IEEE 802.15.4) Standard". In: *International Journal of Engineering Research & Technology (IJERT)* 2.4 (2014), pp. 141–145. URL: https://www.academia.edu/52492484/A_Review_Paper_on_Zigbee_IEEE_802_15_4_Standard.

[14] B. Hammi et al. "Survey on Smart Homes: Vulnerabilities, Risks, and Countermeasures". In: *Computers & Security* 117 (2022), p. 102677. DOI: https://doi.org/10.1016/j.cose.2022.102677.

[15] N.U. Huda et al. "Experts and Intelligent Systems for Smart Homes' Transformation to Sustainable Smart Cities: A Comprehensive Review". In: *Expert Systems with Applications* 238 (2024), p. 122380. DOI: 10.1016/j.eswa.2023.122380.

[16] L.R. Hudson et al. "Remote Physiological Monitoring: Clinical, Financial, and Behavioral Outcomes in a Heart Failure Population". In: *Dis Manag* 8 (2005), pp. 372–381. DOI: 10.1089/dis.2005.8.372.

[17] S. M. Riazul Islam et al. "The Internet of Things for Health Care: A Comprehensive Survey". In: *IEEE Access* 3 (2015), pp. 678–708. DOI: 10.1109/ACCESS.2015.2437951.

[18] Hussain Kazmi, Fahad Mehmood, and Manar Amayri. "Smart Home Futures: Algorithmic Challenges and Opportunities". In: *14th International Symposium on Pervasive Systems, Algorithms and Networks & 11th International Conference on Frontier of Computer Science and Technology & Third International Symposium of Creative Computing (ISPAN-FCST-ISCC)*. 2017, pp. 441–448. DOI: 10.1109/ISPAN-FCST-ISCC.2017.60.

[19] Hakpyeong Kim et al. "A Systematic Review of the Smart Energy Conservation System: From Smart Homes to Sustainable Smart Cities". In: *Renewable and Sustainable Energy Reviews* 140 (2021), p. 110755. DOI: 10.1016/j.rser.2021.110755.

[20] Sachin Kumar, Prayag Tiwari, and Mikhail Zymbler. "Internet of Things Is a Revolutionary Approach for Future Technology Enhancement: A Review". In: *Journal of Big Data* 6.1 (2019), p. 111. DOI: 10.1186/s40537-019-0268-2.

[21] In Lee and Kyoochun Lee. "The Internet of Things (IoT): Applications, Investments, and Challenges for Enterprises". In: *Business Horizons* 58.4 (2015), pp. 431–440. DOI: https://doi.org/10.1016/j.bushor.2015.03.008.

[22] S.S. Mahadik, P.M. Pawar, and Raja Muthalagu. "Heterogeneous IoT (HetIoT) Security: Techniques, Challenges and Open Issues". In: *Multimedia Tools and Applications* (2023), pp. 1–42. DOI: 10.1007/s11042-023-16715-w.

[23] Yuyi Mao et al. "A Survey on Mobile Edge Computing: The Communication Perspective". In: *IEEE communications surveys & tutorials* 19.4 (2017), pp. 2322–2358. DOI: 10.1109/COMST.2017.2745201.

[24] Tony Mariotti. *Smart Home Statistics*. https://www.rubyhome.com/blog/smart-home-stats (accessed: 23.05.2024). 2023.

[25] Kholoud Maswadi, Norjihan Binti Abdul Ghani, and Suraya Binti Hamid. "Systematic Literature Review of Smart Home Monitoring Technologies Based on IoT for the Elderly". In: *IEEE access* 8.39 (2020), pp. 92244–92261. DOI: 10.1109/ACCESS.2020.2992727.

[26] Du Miao et al. "Big Data Privacy Preserving in Multi-Access Edge Computing for Heterogeneous Internet of Things". In: *IEEE Communications Magazine* 56.8 (2018), pp. 62–67. DOI: 10.1109/MCOM.2018.1701148.

[27] Dragos Mocrii, Yuxiang Chen, and Petr Musilek. "IoT-Based Smart Homes: A Review of System Architecture, Software, Communications, Privacy and Security". In: *Internet of Things* 1–2 (2018), pp. 81–98. DOI: 10.1016/j.iot.2018.08.009.

[28] Seliem Mohamed, Khalid Elgazzar, and Kasem Khalil. "Towards Privacy Preserving IoT Environments: A Survey". In: *Wireless Communications and Mobile Computing* (2018), pp. 1–15. DOI: 10.1155/2018/1032761.

[29] M. Noura, M. Atiquzzaman, and M. Gaedke. "Interoperability in Internet of Things: Taxonomies and Open Challenges". In: *Mobile Network Applications* 24 (2019), pp. 796–809. DOI: 10.1007/s11036-018-1089-9.

[30] Antonio Petrosino et al. "Light Fidelity for Internet of Things: A Survey". In: *Optical Switching and Networking* 48 (2023), p. 100732. DOI: 10.1016/j.osn.2023.100732.

[31] Pawani Porambage et al. "Survey on Multi-Access Edge Computing for Internet of Things Realization". In: *IEEE Communications Surveys & Tutorials* 20.4 (2018), pp. 2961–2991. DOI: 10.1109/COMST.2018.2849509.

[32] Amjad Qashlan et al. "Privacy-Preserving Mechanism in Smart Home Using Blockchain". In: *IEEE Access* 9 (2021), pp. 103651–103669. DOI: 10.1109/ACCESS.2021.3098795.

[33] Joao Carlos Resende and Ricardo Chaves. "Compact Dual Block AES Core on FPGA for CCM Protocol". In: *25th International Conference on Field Programmable Logic and Applications (FPL)*. 2015, pp. 1–8. DOI: 10.1109/FPL.2015.7293948.

[34] Rosslin Robles and Tai-hoon Kim. "Applications, Systems and Methods in Smart Home Technology: A Review". In: *International Journal of Advanced Science and Technology* 15 (2010), pp. 37–48. URL: https://www.academia.edu/6341513/Applications_Systems_and_Methods_in_Smart_Home_Technology_A_Review.

[35] Kollolu Roopha. "A Review on Wide Variety and Heterogeneity of IoT Platforms". In: *The International journal of analytical and experimental modal analysis* 12.1 (2020), pp. 3753–3760. DOI: 10.2139/ssrn.3912454.

[36] Nick Ho-Sam-Sooi, Wolter Pieters, and Maarten Kroesen. "Investigating the Effect of Security and Privacy on IoT Device Purchase Behaviour". In: *Computers & Security* 102 (2021), pp. 102–132. DOI: 10.1016/j.cose.2020.102132.

[37] M.A. Al-Shareeda et al. "Bluetooth Low Energy for Internet of Things: Review, Challenges, and Open Issues". In: *Indonesian Journal of Electrical Engineering and Computer Science* 32.2 (2023), pp. 1182–1189. DOI: 10.11591/ijeecs.v31.i2.pp1182-1189.

[38] Z. Shouran, A. Ashari, and T. Priyambodo. "Internet of Things (IoT) of Smart Home: Privacy and Security". In: *International Journal of Computer Applications* 182.39 (2019), pp. 3–8. DOI: 10.1155/2024/7716956.

[39] Manuel Silverio. *What Is a Smart Device?* https://builtin.com/articles/smart-device (accessed: 15.05.2024). 2023.

[40] Swiss-Confederation. *New Federal Act on Data Protection (nFADP).* https://www.kmu.admin.ch/kmu/en/home/facts-and-trends/digitization/data-protection/new-federal-act-on-data-protection-nfadp.html (accessed: 10.03.2024). 2024.

[41] Jacopo Tosi et al. “Performance Evaluation of Bluetooth Low Energy: A Systematic Review”. In: *Sensors* 17.12 (2017). DOI: 10.3390/s17122898.

[42] Paul Voigt and Axel Von dem Bussche. *The EU General Data Protection Regulation (GDPR). A Practical Guide*. Vol. 1. Springer Cham, 2017. DOI: 10.1007/978-3-319-57959-7.

[43] Frederick W. Vook et al. “5G New Radio: Overview and Performance”. In: *52nd Asilomar Conference on Signals, Systems, and Computers*. 2018, pp. 1247–1251. DOI: 10.1109/ACSSC.2018.8645228.

[44] Ming Wang et al. “An IoT-Based Appliance Control System for Smart Homes”. In: *Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*. 2013, pp. 744–747. DOI: 10.1109/ICICIP.2013.6568171.

[45] Li Da Xu, Wu He, and Shancang Li. “Internet of Things in Industries: A Survey”. In: *IEEE Transactions on Industrial Informatics* 10.4 (2014), pp. 2233–2243. DOI: 10.1109/TII.2014.2300753.

[46] Ping Zhang, Mimoza Durresi, and Arjan Durresi. “Multi-Access Edge Computing Aided Mobility for Privacy Protection in Internet of Things”. In: *Computing* 101 (2019), pp. 729–742. DOI: 10.1007/s00607-018-0639-0.

[47] Jun Zhou et al. “Security and Privacy for Cloud-Based IoT: Challenges”. In: *IEEE Communications Magazine* 55.1 (2017), pp. 26–33. DOI: 10.1109/MCOM.2017.1600363CM.

[48] Andrew Zignani and Stephanie Tomsett. “Ultra-Wideband (UWB) for the IoT: A Fine Ranging Revolution”. In: *ABI Research* (2021), pp. 1–23. URL: https://www.allaboutcircuits.com/uploads/articles/UWBWP.pdf.

[49] Alireza Zohourian et al. “IoT Zigbee Device Security: A Comprehensive Review”. In: *Internet of Things* 22 (2023), p. 100791. DOI: 10.1016/j.iot.2023.100791.

Chapter 7

Leveraging Blockchain Technology for Enhanced Financial Services

Cyrill Meier

Blockchain technology, a decentralized digital ledger system, is rapidly transforming the landscape of digital transactions and data management across various industries. This paper explores the fundamental aspects of blockchain technology, including its operational mechanisms, foundational concepts such as immutability, decentralization, and consensus algorithms, and the significant impact it has had on the financial sector. We delve into how blockchain underpins cryptocurrencies like Bitcoin, facilitating secure and transparent transactions without the need for central authority. Further, the paper discusses the broader applications of blockchain beyond financial transactions, such as in legal contracts and healthcare records, emphasizing its potential to enhance transparency, efficiency, and security across different sectors. The introduction of Bitcoin ETFs and the implications of blockchain technology on traditional financial systems, particularly in addressing challenges related to scalability, security, and regulatory compliance, are also examined. By analyzing these elements, this study highlights the revolutionary potential of blockchain to redefine global transaction mechanisms and suggests future directions for its development and integration into mainstream economic systems.

Contents

7.1 Introduction to Blockchain Technology

Blockchain technology, fundamentally a decentralized digital ledger, records transactions across numerous computers in a way that the registered transactions cannot be altered retroactively. This technology underpins cryptocurrencies like Bitcoin, providing a robust architecture that maintains a secure and transparent transaction record. Each 'block' in the chain comprises a number of transactions, and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant's ledger. The decentralization aspect of blockchain is crucial, as it ensures that no single entity has control over the entire chain, thereby enhancing security and integrity. The addition of blocks to the blockchain involves a process known as mining, which requires solving complex cryptographic puzzles. Once solved, the block is added to the chain, a move that is immediately visible to all participants and is secured by the collective agreement of all nodes in the network, thus providing trust and security in a decentralized manner. [23]

Blockchain technology extends far beyond just the financial sector, despite its significant impact there. Originally, a blockchain comprised linked blocks of cryptocurrency transactions, a novel concept that quickly garnered substantial interest, particularly within the FinTech community. Conceptualized by Satoshi Nakamoto in 2008, blockchain technology was first applied in a practical way to timestamp digital documents without the need for a central authority. Each block in the chain securely contains transaction data, a timestamp, and a cryptographic hash of the previous block, linking them securely. This design employs a method similar to Hashcash, allowing blocks to be added sequentially and securely without requiring a trusted intermediary. [34] [26]

The use of blockchain as the foundational technology for Bitcoin and other cryptocurrencies has drawn widespread attention. It is increasingly regarded as a revolutionary framework for conducting and recording transactions globally. This rapid evolution of blockchain has prompted a shift in how businesses operate, driving innovations that promise to transform various industry sectors.

One of the most significant developments in this realm is the introduction of Bitcoin Exchange-Traded Funds (ETFs). An ETF is a type of security that tracks an index, commodity, bonds, or a basket of assets like an index fund but trades like a stock on an exchange. Bitcoin ETFs represent a breakthrough, offering investors a more traditional way to gain exposure to Bitcoin without the complexities of managing actual cryptocurrency holdings. [8]

Blockchain technology is distinguished by several foundational concepts that ensure its robustness, security, and efficiency. These key concepts are pivotal to its function and widespread application across various industries. Here, we explore these critical elements in detail: [34]

Immutability: One of the cornerstone features of blockchain technology is its immutability. Each block within the blockchain is linked to the previous one via a hash pointer, which contains the hash of the data in the previous block. This structure inherently makes it tamper-proof. Hash functions are designed to be one-way and collision-resistant, meaning that once data has been entered into the blockchain, it cannot be altered without redoing all subsequent blocks. This immutable record-keeping is crucial for applications where the integrity of the data is paramount, such as financial transactions, legal contracts, and medical records. [15]

No Single Point of Failure: Unlike traditional centralized databases, blockchain operates on a peer-to-peer (P2P) network architecture. Each node or participant in the network maintains a copy of the entire blockchain, which ensures that there is no single point of failure. Even if one or more nodes experience a malfunction or are attacked, the overall system continues to function effectively without any data loss. This decentralized nature not only enhances the robustness of the system but also distributes trust among all participants. [32]

Consensus Algorithms: Trust among decentralized nodes is achieved through consensus algorithms, which are the rules by which the nodes agree on the state of the blockchain. Since nodes do not necessarily trust each other, they rely on a consensus mechanism to agree on ledger updates. The most widely adopted consensus algorithm in blockchain is Proof of Work (PoW), utilized by networks like Bitcoin. Other popular algorithms include Proof of Stake (PoS) and Practical Byzantine Fault Tolerance (PBFT). Each algorithm has its own mechanism to ensure that all transactions are agreed upon fairly, without the need for a central authority, thereby maintaining the integrity and security of the network. [3], [21], [11]

Data Transparency: Blockchain provides an unprecedented level of transparency. Every transaction on the blockchain is visible to all participants and cannot be changed once confirmed. This transparency ensures that all transactions are verifiable and traceable. In financial contexts, this means that money flows can be tracked and audited in real-time. In supply chains, this visibility helps verify the authenticity of the claimed goods at every point in their journey. Together, these key concepts not only define the operation of blockchain technology but also contribute to its strength as a disruptive technological force in various sectors. By leveraging immutability, decentralization, consensus mechanisms, and transparency, blockchain provides a secure, efficient, and transparent way for information exchange and value transfer in a digital world. [18]

7.2 Understanding the Basics of Financial Services

The financial services industry plays a critical role in the global economy, facilitating transactions, investments, and the management of financial risk. This sector includes a diverse range of services, such as banking, insurance, investments, and real estate, each essential to the economic infrastructure. [1]

Traditional financial systems are centralized, where crucial operations like clearance, settlements, and record-keeping are managed by central entities such as banks and clearinghouses. This centralized structure introduces inefficiencies, making financial transactions slower and more costly, particularly evident in international money transfers governed by the traditional ”SWIFT” system. Centralization also raises significant risks, including fraud, errors, and operational risks due to the extensive human involvement and the intricate nature of global financial regulations.

Moreover, these systems often struggle with accessibility issues, failing to adequately serve underbanked populations, especially in less developed regions. The dependency on physical infrastructure and centralized operations limits service availability and drives up costs for consumers. Additionally, traditional financial systems typically suffer from a lack of transparency, which can lead to corruption and mismanagement, undermining trust in financial institutions. [28]

7.3 The Role of Blockchain in Financial Services

Blockchain technology introduces several transformative advantages for the financial services industry. First, its decentralized nature significantly reduces the reliance on central authorities or intermediaries, leading to lower transaction costs and faster processing times. The inherent design of blockchain also enhances security and trust, as each transaction is encrypted and linked to the previous transaction, making it nearly impossible to alter historical data without detection across the entire network. Furthermore, blockchain technology increases transparency, allowing all parties in the network to view transaction histories, thus reducing the potential for fraud and increasing accountability in financial operations. [23]

Identity Verification

Blockchain technology significantly enhances the Know Your Customer (KYC) process for financial institutions. It streamlines operations by providing a secure, immutable, and transparent framework that enhances the accuracy of identity verification while protecting sensitive personal data. This leads to improved compliance with regulatory requirements and reduced risk of data breaches. [13]

Fraud Reduction

The inherent transparency and immutability of blockchain are instrumental in reducing the incidence of financial fraud. By making every transaction traceable and unalterable once recorded on the blockchain, it becomes significantly more challenging for fraudulent activities to occur unnoticed. [13]

Smart Contracts

Blockchain facilitates the use of smart contracts, which are self-executing contracts with the terms of the agreement directly written into lines of code. These contracts automatically enforce and execute contractual obligations without the need for human intervention, thereby saving time, reducing disputes, and increasing efficiency. [13]

Peer-to-Peer (P2P) Lending

In the realm of syndicated loans, where multiple lenders share a single loan, blockchain can simplify and enhance the efficiency of the agreement and settlement processes. The technology provides a transparent platform that allows all parties to monitor and verify terms and transactions, streamlining the overall process. [13]

Payments

Blockchain enables significantly faster and more cost-effective payment processing compared to traditional systems. By facilitating peer-to-peer transactions without the need for intermediaries, blockchain reduces transaction fees and minimizes processing delays, particularly in cross-border transactions. This technology supports a seamless transfer of value across borders, bypassing the complexities and costs associated with multiple intermediary banks and currency exchanges [25].

Clearing and Settlement

Traditional clearing and settlement processes can be cumbersome, requiring several days to complete due to the need for manual reconciliation among various parties. Blockchain technology streamlines this process by providing a single, immutable ledger that is accessible to all transaction parties. This feature allows for almost instantaneous settlements, reducing counterparty risks and significantly freeing up capital that would otherwise be tied up during the settlement period [17].

Trade Finance

Trade finance is another area where blockchain brings considerable improvements. The traditional reliance on paper-based systems and manual handling in trade finance introduces delays and potential for errors and fraud. Blockchain offers a digitized, secure, and transparent network where all parties-banks, traders, and other intermediaries-can interact with the assurance that the data is accurate and immutable. This technology facilitates faster validations, and the automated execution of agreements through smart contracts ensures compliance and swift processing of trade documentation [4].

Regulatory Compliance

Blockchain substantially aids in meeting regulatory compliance requirements, particularly in sectors such as anti-money laundering (AML) and know your customer (KYC). The transparency of blockchain ensures that all transactions are traceable and immutable, allowing financial institutions to monitor and report suspicious activities more efficiently. Moreover, the shared ledger in a blockchain can be accessed by regulators in real-time, facilitating better monitoring and enforcement of regulatory compliance [10].

7.4 Regulatory Landscape and Challenges

Privacy and Security Concerns

With transparency being one of the primary purposes of blockchains, privacy emerges as a critical concern, especially in the FinTech sector. Implementing blockchain technology must comply with stringent data protection regulations such as the General Data Protection Regulation (GDPR) [19], the Personal Information Protection and Electronic Documents Act (PIPEDA) [9], and the California Consumer Privacy Act (CCPA) [29].

These data protection laws significantly impact the adoption of blockchain within FinTech. For instance, the immutability of blockchain, which is one of its core strengths, conflicts with GDPR's right to be forgotten. This right allows users to withdraw and delete their transactions and personal information, including any encrypted data [19].

One possible solution to address these privacy concerns is utilizing off-chain storage for personal data. In this approach, personal data is stored and maintained off the blockchain, and only a hash of the data is recorded on the ledger. While this method helps comply with privacy regulations, it also reduces the transparency benefit, which is a fundamental attribute of blockchain technology.

Current Regulations and Compliance Issues

With the increasing adoption of blockchain technology globally, countries such as Australia, the US, South Korea, Switzerland, China, the UK, Japan, Singapore, Hong Kong, and Canada are intensifying efforts to regulate blockchain to prevent fraud and other illegal activities that could harm consumer interests and market stability [36]. Regulatory uncertainties pose several challenges as described by Interviewee A: "The technical challenge of Blockchain is that no matter how perfect the Blockchain technology is, it cannot guarantee the authenticity of offline data. The data in question will be permanently recorded on the Blockchain if there is a problem with the data source. Since Blockchain is decentralized, without the supervision of laws and personnel, and it is difficult to change records on the chain, all of these will cause some problems."

Furthermore, some governments view cryptocurrencies as illegal, with Bitcoin being unrestricted in only about 110 countries [33]. This regulatory gap is due to the novelty of the asset class, which has led to a lack of adapted policies by governments and banks. This situation becomes problematic in instances of fraud, bankruptcy, and other failures, especially for companies operating across multiple jurisdictions [22]. The uncertain taxation status and trading rules of cryptocurrencies like Bitcoin could change abruptly, posing significant risks.

Additionally, the absence of robust regulation can lead to market manipulation by a small group of crypto owners. Nguyen (2016) highlighted that the lack of legal and regulatory frameworks on Bitcoin and cryptocurrency significantly hinders the full application of blockchain technology. Nguyen stated, “We are supposed to pay attention to the legitimacy of Blockchain. Although there are no specific regulations on Blockchain until now, relevant laws might be introduced once some new products of Blockchain appear. The award method is one of the intrinsic properties of Blockchain, so how to define the nature of these rewards, whether these conducts violate the law, all of these are needed to be discussed” [27].

Regulatory Challenges in Adopting Blockchain

One of the primary regulatory challenges in adopting blockchain within financial services is the technology’s decentralized nature, which does not fit neatly into traditional regulatory frameworks designed for centralized financial systems. Regulatory bodies face difficulties in applying old rules to new paradigms, such as dealing with the anonymity of cryptocurrency transactions, which poses challenges for anti-money laundering (AML) and combating the financing of terrorism (CFT). Moreover, the cross-jurisdictional nature of blockchain can complicate the enforcement of regulations when multiple countries with differing laws are involved [22] [27] .

7.5 Implementing Blockchain in Financial Institutions

This section explores the Blockchain Fit Assessment Framework as applied within the banking sector, focusing on identifying processes where blockchain technology can provide significant enhancements. The framework evaluates potential blockchain integration across various banking functions by assessing critical factors such as intermediaries, transparency needs, information storage, manual processing, trust, documentation, and time sensitivity. This systematic approach aims to determine the appropriateness of blockchain solutions for specific banking processes, ensuring that the technology’s deployment aligns with strategic operational improvements and addresses existing pain points effectively. The second part of this section, will be focusing on a concrete Case Study in the Indian Banking Sector.

Blockchain Fit Assessment Framework

In an era where efficiency and security are paramount in banking, blockchain technology presents a transformative approach to redesign traditional banking systems. The Blockchain Fit Assessment Framework serves as a critical tool for banks, helping them evaluate and select optimal processes for blockchain implementation. It identifies areas where blockchain can reduce costs, enhance speed, improve transparency, and bolster security. [35]

The Framework's Components

The Blockchain Fit Assessment Framework consists of several components that collectively determine the suitability of blockchain for a banking process:

- **Intermediary Needs**: Examines whether the process involves intermediaries who add complexity and cost.
- **Transparency Requirements**: Considers whether increased transparency could enhance process integrity and stakeholder trust.
- **Information Storage**: Assesses the efficiency gains from decentralized storage.
- **Manual Processing**: Identifies processes that are labor-intensive and prone to errors, making them ideal candidates for automation through blockchain.
- **Trust Factors**: Evaluates the level of trust among participants and the potential for blockchain to enhance it.
- **Documentation**: Looks at whether the process involves extensive paperwork that blockchain could digitize and streamline.
- **Time Sensitivity**: Considers if the timeliness of the process is critical, which blockchain could improve through real-time updates. [35]

Application of the Framework

Applying the Blockchain Fit Assessment Framework involves a detailed analysis of each component within a banking process. For example, in cross-border payments, the framework would analyze the high costs associated with intermediaries, the benefits of increased transaction transparency, and the impact of real-time settlements on transaction efficiency. Each component is scored based on its alignment with blockchain capabilities, guiding decision-makers in pinpointing processes that would benefit most from blockchain integration. [35]

Benefits and Impacts

The implementation of blockchain, as suggested by the framework, offers numerous benefits:

- **Reduced Costs**: By eliminating intermediaries and reducing manual processing.
- **Enhanced Security**: Through immutable and transparent record-keeping.
- **Increased Efficiency**: Via automated processes and faster settlements.
- **Improved Compliance**: With enhanced data integrity and audit trails. [35]

Case Study in the Indian Banking Sector

Steps to Implement Blockchain Solutions

The Indian banking sector is progressively adopting blockchain technology to enhance efficiency and security. The implementation process involves a sequence of strategic steps:

1. Formation of internal specialized teams tasked with researching blockchain technology and its potential impact on banking operations.
2. Making informed strategic investment decisions by studying the implementation of blockchain in banks across the globe.
3. Selecting an appropriate blockchain ecosystem that aligns with the bank's scalability needs and service offerings.
4. Ensuring that the integration of blockchain with existing banking infrastructure does not compromise data security.
5. Initiating pilot projects to explore the tangible benefits of blockchain and developing strategies for data privacy and security.

These steps constitute a framework that Indian banks are using to navigate the transition towards blockchain-enabled banking services, balancing innovation with prudence.

Considerations for Blockchain Integration in the Indian Banking Sector

As the Indian banking sector integrates blockchain technology, several key considerations emerge:

1. The interoperability and standardization of blockchain systems are crucial for seamless integration and are hindered by the lack of common international standards.
2. The legal and regulatory framework must be clearly defined, especially for transactions that span multiple jurisdictions.
3. Operational feasibility is vital, requiring the capability to assimilate blockchain within the existing systems and practices among various stakeholders.
4. Ensuring multi-level security is paramount in protecting the blockchain infrastructure against cyber threats.
5. Regulatory bodies must be fully cooperative and involved to enable the adoption of blockchain technology within the banking ecosystem.
6. Cost distribution strategies should be devised in collaboration with partners to ensure long-term sustainability.

7. Data privacy considerations dictate that only pertinent and secure transaction information is shared on the blockchain, respecting customer privacy and data protection regulations.

These considerations play a pivotal role in the successful integration of blockchain technology, dictating the operational effectiveness and adoption rate within the banking sector. [30], [37]

7.6 Future Trends and Opportunities

CBDCs are issued as legal tender by a country's central bank, incorporating features designed to facilitate efficient and secure financial operations:

- **Centralized Issuance:** Governed by the central bank, CBDCs carry intrinsic value derived from national economic policies.
- **Transferability:** They function as a medium of exchange for economic activities, maintaining a zero-sum game to ensure one entity's gain is another's loss.
- **Storability:** CBDC transactions are securely recorded in digital formats, accessible via electronic devices for effective payment management.
- **Offline Transactions:** CBDCs support offline functionalities, catering to individuals without constant internet access or sophisticated devices, similar to traditional credit and debit systems.
- **Exchangeability:** Facilitates easy conversion between CBDCs and other digital or fiat currencies, enhancing liquidity and reducing exchange times. [20]

Various models have been proposed to address the unique requirements of CBDCs:

- **Permissioned Blockchain Models:** Sun et al. suggested using permissioned blockchain managed by central and commercial banks, enhancing security and mitigating risks like double-spending.
- **Hybrid Digital Currency Schemes:** Zhang et al. introduced a model combining Unspent Transaction Output (UTXO) and Account schemes to optimize transaction efficiency and data storage. [38]

Challenges in Implementing CBDCs

Implementing CBDCs involves several challenges that need to be meticulously addressed:

- **Security Concerns:** Ensuring the safety of private keys and preventing unauthorized transactions is paramount.

- **Scalability Issues:** The public key infrastructure (PKI) and other technologies must handle vast numbers of transactions without performance degradation.
- **Regulatory Compliance:** CBDCs must comply with KYC and anti-money laundering laws, posing challenges for transaction auditing and data privacy.
- **Technological Overheads:** Advanced schemes like Zero-Knowledge Proofs (ZKPs) are required to audit transactions while preserving privacy, adding complexity to the CBDC systems. [7], [12]

Current Status of CBDCs

As the financial landscape evolves, several countries have made significant strides in CBDC development. This section details the progress in various regions.

Switzerland

The Swiss National Bank (SNB) has been conducting research and pilot projects to understand and evaluate the implications of introducing a CBDC.

One notable initiative is the ”Project Helvetia,” a collaboration between the SNB, the Bank for International Settlements (BIS), and the Swiss financial services company SIX. This project explores the integration of digital central bank money into a distributed ledger technology (DLT) platform. The project aims to examine the legal, technical, and policy aspects of processing tokenized assets with both wholesale and potentially retail CBDCs.

Project Helvetia has successfully demonstrated the feasibility of integrating digital central bank money into existing banking systems and the DLT platform operated by SIX. The experiments conducted under this project have focused on wholesale CBDC applications, meaning the digital currency is used for large-scale transactions between financial institutions rather than for everyday consumer use.

Switzerland’s approach has been somewhat cautious and focused on thorough research before making any decisions about wide-scale implementation. The SNB has emphasized that its exploration of CBDCs does not necessarily mean that a Swiss CBDC will be issued imminently but is part of its ongoing efforts to stay at the forefront of technological developments in the financial sector. [14] [2]

7.7 Risks and Mitigation Strategies

Scalability Issues

The scalability of blockchain technology becomes increasingly challenging as transaction volumes grow. Zheng et al. [39] note that the blockchain can become unwieldy with the addition of numerous transactions. Marr [24] elaborates that the complexity, encrypted nature, and distributed architecture of blockchain transactions contribute to delays in their processing times.

Ethereum, an open-source, public, blockchain-based computing platform, also generates a cryptocurrency known as Ether. According to Chen et al. [16], the platform supports over one million smart contracts and has become a hub for numerous developers and entrepreneurs to launch new projects and startups.

In terms of transaction processing speed, it significant dispares when compared to traditional payment systems: Visa processes approximately 24,000 transactions per second, PayPal manages 193, whereas Ethereum and Bitcoin can handle only about 20 transactions per second. This limitation is critical, as it impedes the blockchain's capacity to process large volumes of transactions swiftly. The primary constraint lies in the limited capacity of blockchain blocks, which leads to delays in processing smaller transactions, as miners often prioritize transactions that include higher fees [5].

Solana's innovation in blockchain scalability is evidenced by its adoption of the Proof of History (PoH) consensus, which enables high throughput without sacrificing security. In a study that collected and analyzed data over a two-month period, the Solana blockchain demonstrated an average transactions' throughput of approximately 2812 transactions per second (TPS). This significantly surpasses the capabilities of more established blockchains like Bitcoin and Ethereum. Furthermore, the study observed that transaction fees on Solana are substantially lower than those on comparable blockchains, making it an economically attractive option for users and developers alike. [31]

7.8 Public Perception and Trust

Public perception of blockchain technology is tinged with significant concerns that could hinder its broader adoption. Notably, apprehensions regarding the legality and regulatory environment surrounding blockchain and cryptocurrencies play a pivotal role in shaping public sentiment. These concerns are critical as they influence trust and acceptance among potential users and investors, affecting the overall market stability and growth potential of blockchain applications. Furthermore, the public's worry about the volatility of cryptocurrencies and the security of blockchain transactions underscores the need for robust, clear regulatory frameworks and advanced security solutions to foster a safe, stable environment for blockchain operations.

The results from the sentiment analysis reveal a nuanced landscape of public opinion across different regions. From the bar graph presented in Figure 1.1, it is evident that Sweden leads with the highest positive public sentiment towards cryptocurrency, which could be attributed to its supportive regulatory environment and a high degree of technological integration in society. In contrast, the UK exhibits more cautious and less positive attitudes, likely influenced by ongoing debates about cryptocurrency regulations and economic implications. These regional variations in public sentiment underscore the importance of localized approaches to policy-making and community engagement to promote the adoption of blockchain technology. [6]

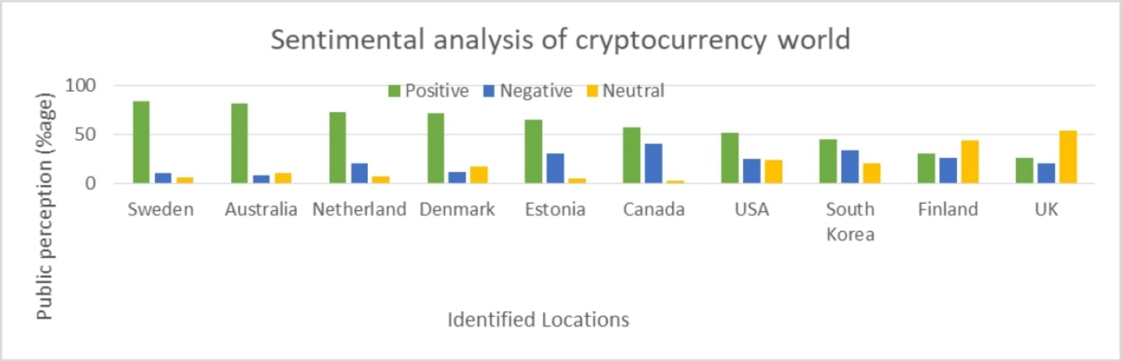


Figure 7.1: Public perception (positive, negative and neutral) in top identified locations

7.9 Conclusion

This paper has critically examined the transformative potential of blockchain technology in redefining the mechanisms of transactions and data management across various sectors. Through an in-depth analysis of blockchain's foundational concepts-immutability, decentralization, consensus algorithms, and transparency-we have demonstrated how it establishes a robust framework for secure and efficient digital transactions. Notably, blockchain's significant impact on the financial sector has been highlighted, illustrating its role in enhancing the transparency, efficiency, and security of financial transactions. This includes its application in innovative financial instruments such as Bitcoin ETFs, which bridge the gap between traditional investment mechanisms and the digital currency space.

The exploration of blockchain's applications has extended beyond financial transactions to include critical areas such as legal documentation, healthcare records, and identity verification, showcasing its versatility and wide-ranging implications. Furthermore, the discussion on the challenges of scalability and regulatory compliance underscores the ongoing need for development and adaptation in blockchain technology.

As blockchain technology continues to evolve, it holds the promise to radically alter not just financial systems but also the way in which transparent and secure data management is conducted globally. Future research should focus on addressing the scalability challenges, enhancing the interoperability among diverse blockchain systems, and developing more inclusive regulatory frameworks that can keep pace with the rapid advancements in blockchain applications.

In conclusion, blockchain technology not only offers a powerful tool for managing transactions and data but also serves as a catalyst for innovation across numerous fields. By continuing to harness its potential responsibly, blockchain can provide a foundational technology that supports a more efficient, transparent, and secure global information exchange.

Bibliography

[1] O. Ali, Clutterbuck M. Ally, and Y. Dwivedi. "The state of play of blockchain technology in the financial services sector: A systematic literature review". In: *International Journal of Information Management* 54.102199 (2020), pp. 1–13. URL: https://komodoplatform.com/en/academy/blockchain-technology-types/.

[2] SIX Group Bank for International Settlements and Swiss National Bank. *Project Helvetia Phase II: Settling tokenised assets in wholesale CBDC*. Jan. 2022. URL: https://www.bis.org/publ/othp45.htm.

[3] S. Bano et al. "Consensus in the age of blockchains". 2017.

[4] J. K. Bartholomew. "Application of Blockchain Technology in the Manufacturing Industry". In: *Journal of Industrial Technology* 34.3 (2018), pp. 22–27.

[5] B. Biais et al. "The Blockchain folk theorem". In: *Rev. Financ. Stud.* 32.5 (2019), pp. 1662–1715.

[6] S. Bibi, S. Hussain, and M. I. Faisal. "Public Perception Based Recommendation System for Cryptocurrency". In: *Proceedings of the 2019 16th International Bhurban Conference on Applied Sciences & Technology (IBCAST)*. Pakistan: Islamabad, Jan. 2019, pp. 661–665.

[7] W. Bolt, V. Lubbersen, and P. Wierts. *Getting the balance right: Crypto, stablecoin and CBDC*. Working Paper 736. De Nederlandsche Bank, Jan. 2022. DOI: 10.2139/ssrn.4014319.

[8] Alessio Brini and Jimmie Lenz. *Bitcoin ETFs: Measuring the Performance of This New Market Niche*. July 2022. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4157711.

[9] The Privacy Commissioner of Canada. *The Personal Information Protection and Electronic Documents Act (PIPEDA)*. Aug. 1, 2022. 2022. URL: https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/.

[10] M. Casey and P. Vigna. "The Age of Cryptocurrency: How Bitcoin and the Blockchain Are Challenging the Global Economic Order". In: *Digital Currency Initiative, MIT Media Lab* (2015).

[11] A. Castor. "A (short) guide to blockchain consensus protocols (2017)". In: (2017).

[12] D. Chang, H. Wang, and Z. Wu. "Maximum principle for non-zero sum differential games of BSDEs involving impulse controls". In: *Proc. 32nd Chin. Control Conf. 2013*, 2013, pp. 1564–1569.

[13] V. Chang et al. *How Blockchain can impact financial services, The overview, challenges and recommendations from expert interviewees*. School of Computing, Engineering and Digital Technologies, Teesside University, 2022.

[14] D. Chaum, C. Grothoff, and T. Moser. *How to issue a central bank digital currency.* Working Paper 3/2021. Zurich, Switzerland. Swiss National Bank, Mar. 2021.

[15] F. Chen et al. "Secure scheme against compromised hash in proof-of-work blockchain". In: Proc. Int. Conf. Netw. Syst. Secur., 2018, 2018, pp. 1–15.

[16] W. Chen et al. "Detecting ponzi schemes on ethereum: towards healthier blockchain technology". In: *Proceedings of the World Wide Web Conference on World Wide Web.* 2018, pp. 1409–1418. DOI: `10.1145/3178876.3186046`.

[17] K. Croman et al. "On Scaling Decentralized Blockchains". In: *Proc. 3rd Workshop on Bitcoin and Blockchain Research.* 2016.

[18] *Four types of blockchain technology.* 2022.

[19] Eu Gdpr. *What is GDPR, the EU's New Data Protection Law?* Aug. 1, 2022. 2022. URL: `https://gdpr.eu/what-is-gdpr/`.

[20] X. Han, Y. Yuan, and F.-Y. Wang. "A blockchain-based framework for central bank digital currency". In: *Proc. IEEE Int. Conf. Serv. Operations Logistics, Inform.* 2019, 2019, pp. 263–268.

[21] A. Juels and J. Brainard. "Cryptographic countermeasures against connection depletion attacks". In: *U. S. Patent* 7197639, 27 (Mar. 2007).

[22] R. Lewis, J. W. McPartland, and R. Ranjan. "Blockchain and financial market innovation". In: *Economic Perspectives* 41 (2017), pp. 1–17.

[23] *Markets and markets report.* Mar. 2023. URL: `https://www.marketsandmarkets.com/Market-Reports/blockchain-technology-market-90100890.html`.

[24] B. Marr. *The 5 Big Problems with Blockchain Everyone Should Be Aware Of.* Available: Forbes, Feb. 19. 2018. URL: `https://www.forbes.com/sites/bernardmarr/2018/02/19/the-5-big-problems-with-blockchain-everyone-should-be-aware-of/?sh=4f01c45d1670`.

[25] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System.* 2008.

[26] Arvind Narayanan et al. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction.* Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton Univsersity Press, 2016. URL: `https://www.marketsandmarkets.com/Market-Reports/blockchain-technology-market-90100890.html`.

[27] Q. K. Nguyen. "Blockchain - A Financial Technology for Future Sustainable Development". In: *Proceedings of the 3rd International Conference on Green Technology and Sustainable Development (GTSD).* 2016, pp. 51–54. DOI: `10.1109/GTSD.2016.22`.

[28] A. Norta, B. Leiding, and A. Lane. "Lowering financial inclusion barriers with a blockchain-based capital transfer system". In: *Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS).* Paris, Apr. 2019, pp. 319–324.

[29] State of. *California Consumer Privacy Act (CCPA).* Aug. 1, 2022. 2022. URL: `https://oag.ca.gov/privacy/ccpa`.

[30] A. Patki and V. Sople. "Indian banking sector: blockchain implementation, challenges and way forward". In: *Journal of Banking and Financial Technology* 4 (2020), pp. 42786–020. DOI: `10.1007/s42786-020-00019-w`.

[31] G. A. Pierro and R. Tonelli. "Can Solana be the Solution to the Blockchain Scalability Problem?" In: *Proc. IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER).* 2022, pp. 1219–1226. DOI: `10.1109/SANER53432.2022.00144`.

[32] B. Pillai et al. “Crossblockchain technology: Integration framework and security assumptions”. In: *IEEE Access* 10 (2022), pp. 41239–41259.

[33] D. Price. *5 Big Blockchain Issues: Security, Privacy, Legal, Regulatory, and Ethical.* Oct. 3, 2018. 2018. URL: `https://blocksdecoded.com/Blockchain-issues-security-privacy-legal-regulatory-ethical/`.

[34] *Report on Survey of Blockchain Technology: Potential to Disrupt All Industries.* 2016. URL: `https://www.chainnode.com/doc/415`.

[35] T. Shah and S. Jani. *Applications of Blockchain Technology in Banking & Finance.* Technical Report. Parul University, Vadodara, India, Feb. 2018. DOI: `10.13140/RG.2.2.35237.96489`.

[36] B. M. Till et al. “From Blockchain technology to global health equity: can cryptocurrencies finance universal health coverage?” In: *BMJ Global Health* 2.4 (2017), e000570. DOI: `10.1136/bmjgh-2017-000570`.

[37] S. Yoo. “Blockchain based financial case analysis and its implications”. In: *Asia Pacific Journal of Innovation and Entrepreneurship* 11.3 (2017), pp. 312–321. DOI: `10.1108/APJIE-12-2017-036`.

[38] J. Zhang et al. “A hybrid model for central bank digital currency based on blockchain”. In: *IEEE Access* 9 (2021), pp. 53589–53601.

[39] Z. Zheng et al. “Blockchain challenges and opportunities: a survey”. In: *Int. J. Web Grid Serv.* 14.4 (2018), pp. 352–375. DOI: `10.1504/IJWGS.2018.095647`.