Privacy-Preserving Systems around Security, Trust and Identity

by

PAVLOS PAPADOPOULOS

A thesis submitted in partial fulfilment of the requirements

of Edinburgh Napier University, for the award of

Doctor of Philosophy

School of Computing

Edinburgh Napier University

SEPTEMBER 2022

Dedication

"Can we build a secure world on top of an insecure one?"

– Bruce Schneier [1]

Author's declaration

I, Pavlos Papadopoulos, confirm that this thesis and the work presented in it are my own achievement.

1. Where I have consulted the published work of others this is always clearly attributed.
2. Where I have quoted from the work of others the source is always given. With the exception of such quotations this thesis is entirely my own work.
3. I have acknowledged all main sources of help.
4. If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself.
5. I have read and understand the penalties associated with Academic Misconduct.
6. I also confirm that I have obtained informed consent from all people I have involved in the work in this thesis following the School's ethical guidelines.

TYPE NAME: PAVLOS PAPADOPOULOS

DATE: FRIDAY, 30 SEPTEMBER 2022

MATRICULATION NO: 40413177

Acknowledgements

I would like to sincerely thank my director of studies Dr Nick Pitropakis, for his thoughtful advice, invaluable and continuous support throughout my PhD. Additionally, I would like to thank my supervisors, Prof William J. Buchanan and Dr Owen Lo, for being there every time I needed to discuss any obstacles arisen throughout this PhD. Special thanks also go to the Blockpass Identity Lab (BIL) for the fruitful discussions we had on the presented topics. Finally, I would like to thank my family for the encouragement and all the support given throughout my studies. Last but not least, I would like to sincerely thank my partner, Ergini, who fully supported me and was there for me throughout this journey.

Abstract

Data has proved to be the most valuable asset in a modern world of rapidly advancing technologies. Companies are trying to maximise their profits by getting valuable insights from collected data about people's trends and behaviour which often can be considered personal and sensitive. Additionally, sophisticated adversaries often target organisations aiming to exfiltrate sensitive data to sell it to third parties or ask for ransom. Hence, the privacy assurance of the individual data producers is a matter of great importance who rely on simply trusting that the services they use took all the necessary countermeasures to protect them.

Distributed ledger technology and its variants can securely store data and preserve its privacy with novel characteristics. Additionally, the concept of self-sovereign identity, which gives the control back to the data subjects, is an expected future step once these approaches mature further. Last but not least, big data analysis typically occurs through machine learning techniques. However, the security of these techniques is often questioned since adversaries aim to exploit them for their benefit.

The aspect of security, privacy and trust is highlighted throughout this thesis which investigates several emerging technologies that aim to protect and analyse sensitive data compared to already existing systems, tools and approaches in terms of security guarantees and performance efficiency.

The contributions of this thesis derive to i) the presentation of a novel distributed ledger infrastructure tailored to the domain name system, ii) the adaptation of this infrastructure to a critical healthcare use case, iii) the development of a novel self-sovereign identity healthcare scenario in which a data scientist analyses sensitive data stored in the premises of three hospitals, through a privacy-preserving machine learning approach, and iv) the thorough investigation of adversarial attacks that aim to exploit machine learning intrusion detection systems by "tricking" them to misclassify carefully crafted inputs such as malware identified as benign.

A significant finding is that the security and privacy of data are often neglected since they do not directly impact people's lives. It is common for the protection and confidentiality of systems, even of critical nature, to be an afterthought, which is considered merely after malicious intents occur. Further, emerging sets of technologies, tools, and approaches built with fundamental security and privacy principles, such as the distributed ledger technology, should be favoured by existing systems that can adopt them without significant changes and compromises. Additionally, it has been presented that the decentralisation of machine learning algorithms through self-sovereign identity technologies that provide novel end-to-end encrypted channels is possible without sacrificing the valuable utility of the original machine learning algorithms.

However, a matter of great importance is that alongside technological advancements, adversaries are becoming more sophisticated in this area and are trying to exploit the aforementioned machine learning approaches and other similar ones for their benefit through various tools and approaches. Adversarial attacks pose a real threat to any machine learning algorithm and artificial intelligence technique, and their detection is challenging and often problematic. Hence, any security professional operating in this domain should consider the impact of these attacks and the protection countermeasures to combat or minimise them.

TABLE OF CONTENTS

DEDICATION **i**

AUTHOR'S DECLARATION **ii**

ACKNOWLEDGEMENTS **iii**

ABSTRACT **iv**

TABLE OF CONTENTS **vi**

LIST OF FIGURES **xii**

1 INTRODUCTION **1**

1.1 Introduction 1

1.2 Blockchain Technology 2

1.2.1 Blockchain Consensus Algorithms 6

1.2.2 Distributed Ledger Technology 8

1.3 Machine Learning 10

1.3.1 Federated Learning 11

1.3.2 Split Neural Networks 12

1.4 Research Design and Motivation 12

1.4.1 Aim and Objectives 13

1.4.2 Contributions and Novelty 14

1.4.3 Research Methodology 15

1.4.4 Publications arising from this work 16
1.5 Overview-Structure of the Thesis 18
2 LITERATURE REVIEW 20
2.1 Introduction 20
2.2 Domain Name System and Attacks 20
2.2.1 Phishing 23
2.3 Passive DNS and Blockchain 24
2.3.1 Challenges in a Blockchain Passive DNS system 28
2.4 Healthcare and Blockchain 28
2.4.1 Challenges in the Healthcare domain 30
2.5 Data Privacy Assurance using Blockchain 32
2.5.1 Internet of Things 33
2.5.2 Big Data 34
2.5.3 Cloud Computing 35
2.6 Intrusion Detection Systems 36
2.6.1 Machine Learning Intrusion Detection Systems 37
2.7 Attacks against Machine Learning algorithms 39
2.7.1 Privacy Attacks on Training Data 39
2.7.1.1 Membership Inference Attacks 39
2.7.1.2 Inference of Training Data 40
2.7.1.3 Model Inversion Attacks 41
2.7.1.4 Model Encoding Attacks 44
2.7.2 Privacy Attacks on ML Models 45
2.7.3 Security Attacks on ML Models 45
2.7.3.1 Adversarial Examples 46
2.7.3.2 Data Poisoning Attacks 46
2.7.3.3 Model Poisoning Attacks 47
2.8 Defences against Machine Learning Attacks 48
2.8.1 Mitigation Techniques for Privacy Attacks on Training Data . . . 49

2.8.1.1 Privacy Engineering 49
2.8.1.2 Privacy-Preserving Record Linkage 50
2.8.1.3 Trusted Execution Environments 50
2.8.1.4 Homomorphic Encryption 52
2.8.1.5 Secure Multi-Party Computation 52
2.8.1.6 Differential Privacy 53
2.8.1.7 MLPrivacyGuard 55
2.8.1.8 Knowledge Distillation 55
2.8.1.9 Anomaly Detection 56
2.8.1.10 Defences Against Model Extraction attacks 56
2.8.2 Mitigation Techniques for Security Attacks on ML Models 57
2.8.2.1 Adversarial Training 57
2.8.2.2 Random Self-Ensemble 58
2.8.2.3 Defences Against Data Poisoning Attacks 58
2.8.2.4 Defences Against Model Poisoning Attacks 59
2.9 Summary 60
2.10 Conclusion 62

3 ENSURING DATA PRIVACY USING BLOCKCHAIN 63

3.1 Introduction 63
3.2 Hyperledger Fabric Private-Permissioned Distributed Ledger Technology 65
3.3 Privacy-Preserving Passive DNS 72
3.3.1 Architecture and Implementation 73
3.3.2 Evaluation and Discussion 77
3.3.2.1 Security Evaluation 77
3.3.2.2 Performance Evaluation 79
3.3.3 Summary 81
3.4 Case Study: A Privacy-Preserving Electronic Health Records management system 83
3.4.1 Architecture and Implementation 85

3.4.2 Evaluation and Discussion 87
3.4.2.1 Security Evaluation 87
3.4.2.2 Performance Evaluation 88
3.4.3 Summary 91
3.5 Conclusion 92

4 DIGITAL IDENTITIES AND PRIVACY-PRESERVING MACHINE LEARNING 94

4.1 Introduction 94
4.2 Self-Sovereign Identity Foundations 95
4.2.1 Decentralised Identifiers 95
4.2.1.1 Decentralised Identifiers Communication Protocol . . 97
4.2.2 Verifiable Credentials 97
4.3 Private and Trusted Federated Machine Learning 100
4.3.1 Architecture and Implementation 100
4.3.2 Evaluation 104
4.3.2.1 Performance Evaluation 104
4.3.2.2 Security Evaluation 107
4.3.3 Summary 111
4.4 Conclusion 112

5 ADVERSARIAL MACHINE LEARNING 113

5.1 Introduction 113
5.2 Adversarial Attacks against Network Intrusion Detection Systems for IoT 114
5.2.1 Architecture and Methodology 115
5.2.1.1 Attacks Composition 116
5.2.2 Experimental Implementation 117
5.2.2.1 Data Preparation 117
5.2.2.2 SVM Model 118
5.2.2.3 ANN Model 119
5.2.2.4 Generating Adversarial Examples 121

SVM Model. 121

ANN Model. 123

5.2.3 Experimental Evaluation 124

5.2.3.1 Evaluating the SVM model 124

5.2.3.2 Evaluating the ANN model 128

5.3 Conclusion 131

6 CONCLUSIONS 133

6.1 Future Work 136

REFERENCES 138

APPENDIX A LIST OF ADDITIONAL PUBLICATIONS 181

APPENDIX B MACHINE LEARNING METRICS 184

APPENDIX C AUTOMATED MACHINE LEARNING ANALYSIS OF PHISHING URLS 185

C.1 Datasets 185

C.2 Analysis 187

C.2.1 Descriptive Features 188

C.2.2 Statistical Features 189

C.3 Results 190

C.4 Discussion 193

C.4.1 Feature Importance 193

C.5 Alerting and Further Improvement 194

C.6 Summary 196

APPENDIX D ADDITIONAL TECHNICAL EXPERIMENTATION RELATED TO DIGITAL IDENTITIES AND PRIVACY-PRESERVING MACHINE LEARNING 197

D.1 Case Study: PyDentity and Aries-Jupyter-Playground 197

D.1.1 Architecture 198

D.1.2 Summary 200

D.2 Federated Machine Learning in Vertically Distributed Datasets 200

D.2.1 Architecture 201

D.2.2 Experiment 202

D.2.3 Summary 203

APPENDIX E PRACTICAL DEFENCES AGAINST MODEL INVERSION ATTACKS FOR SPLIT NEURAL NETWORKS 205

E.1 Architecture 206

E.2 Experimental Evaluation 208

E.3 Summary 211

LIST OF FIGURES

FIGURES **Page**

1.1 Blockchain developments overview [29]. 9

1.2 Overview of the research methodology of this thesis. 17

2.1 Overview of the Domain Name System [66] 21

2.2 Membership Inference attacks overview as described in [183]. 41

2.3 Model Inversion attacks overview, adopted from [188]. An attacker that has access to some target labels, queries the original model and exploits the classification values to reconstruct the rest of the data. 42

2.4 Trojaning attack overview [213]. 49

2.5 Trusted Execution Environments overview 51

2.6 The privacy budget needs to be carefully chosen to balance utility and privacy. Images from [236]. 54

2.7 The PATE design first trains multiple teachers and then one student model on top of the teachers' noised labels in order to achieve a differentially private student model. The figure is from [237]. 55

3.1 Visualisation of a distributed ledger infrastructure using the Hyperledger Explorer. 67

3.2 PRESERVE DNS proof-of-concept implementation architecture for the test data collection [66]. 74

3.3 PRESERVE DNS Hyperledger Fabric Infrastructure [66]. 75

3.4 Read Data transactions overhead [66]. 81

3.5 CPU Usage (%) of Nodes during workflow [66]. 82

3.6 Memory Usage (%) of Nodes during workflow on 1,000, 10,000 and 100,000 DNS Entries [66]. 83

3.7 PREHEALTH overview [67]. 84

3.8 PREHEALTH internal technical architecture [67]. 86

3.9 PREHEALTH Hyperledger Fabric architecture [67]. 87

3.10 Read records transactions overhead [67]. 89

3.11 (**a**) Read queries workflow on 1000 Records. (**b**) Write queries workflow on 1000 Records. (**c**) Read queries workflow on 10,000 Records. (**d**) Write queries workflow on 10,000 Records. (**e**) Read queries workflow on 100,000 Records. (**f**) Write queries workflow on 100,000 Records. CPU Usage (%) of blockchain peers during workflow [67]. 91

4.1 Verifiable Credential Roles [68, 293]. 98

4.2 Healthcare trust model overview [68]. 101

4.3 Networking communication architecture [68]. 103

4.4 CPU, Memory usage and Network use of Docker container agents during workflow using the original federated learning architecture of [47] [68]. . . . 105

4.5 CPU Usage comparison of Docker containers during workflow using the novel federated learning libraries of Papadopoulos et al. [68]. 106

4.6 Traffic through the DIDComm protocol is encrypted [68]. 110

5.1 (**a**) Support Vector Machines confusion matrix without label flipping. (**b**) Support Vector Machines Receiving Operator Characteristic curve without label flipping [69]. 119

5.2 Artificial Neural Networks Design [69]. 120

5.3 (**a**) Binary Artificial Neural Networks confusion matrix without adversarial examples. (**b**) Multi-class Artificial Neural Networks confusion matrix without adversarial examples [69]. 120

5.4 SVM model’s metrics during label flipping activities [69]. 122

5.5 ANN binary classification model's metrics during label flipping activities using FGSM [69]. 125

5.6 ANN five-class classification model's metrics during label flipping activities using FGSM [69]. 126

5.7 Confusion matrices of binary and five-class classification ANN models using FGSM perturbed with 1.0 epsilon [69]. 127

5.8 Comparing recall scores in targeted and non-targeted SVM label flipping [69].128

5.9 Comparing accuracy versus recall scores in binary targeted and non-targeted FGSM adversarial examples [69]. 130

5.10 Comparing accuracy versus recall scores in five-class targeted and non-targeted FGSM adversarial examples [69]. 131

C.1 System Architecture Diagram [43]. 186

C.2 Feature importance comparison graph [43]. 194

C.3 Applying the knowledge of a machine learning model into a specified database file in Splunk Search [43]. 195

C.4 Sample results from the Splunk's search machine learning model fit [43]. . . 195

C.5 Splunk machine learning alert [43]. 196

D.1 Aries-Jupyter-Playground Overview [335]. 199

D.2 Parties and datasets in the conducted experiment [336]. 202

D.3 Computation of intersections [336]. 203

D.4 Train and validation-testing accuracy for an unoptimised dual-headed SplitNN on vertically-partitioned MNIST [336]. 204

E.1 MNIST images R reconstructions from adversary models trained on a different numbers of datapoints [345]. 208

E.2 Accuracies on a validation dataset by classifiers with as a function of the scale of laplacian noise added to intermediate data after training [345]. . . 209

E.3 (**a**) MNIST data reconstructions from an attacker trained on 5,000 MNIST images. (**b**) MNIST data reconstructions from an attacker trained on 5,000 EMNIST images [345]. 210

E.4 (**a**) Model inversion attack on an MNIST classifier using the *noise defence* mechanism. Left-to-right: true image, reconstructions on models with (0, 0.1, 0.5, 1.0) noise scale. (**b**) Model inversion attack on an MNIST classifier using NoPeekNN defence. Left-to-right: true image, reconstruction on models with (0, 0.1, 0.5) NoPeekNN weighting [345]. 211

First Chapter

Introduction

1.1 Introduction

Technology is advancing quickly, and there are continuously new inventions and emerging technologies that aim to improve citizens' quality of life and solve previously unsolvable problems. However, the security of these systems is often neglected or is not the priority, and this enables new routes for exploitation by adversaries. Since the last century, every service has been digitalised, in some way; hence, the attack surface that security engineers should guard has widened. Regarding cyber attacks, adversaries typically have significant economic incentives for breaching the security of specific systems and organisations, and as the technology advances, the tools, approaches, and technologies that adversaries use, evolve as well [2]; hence, the available defences, legislations and regulations should be ever-changing, too.

Privacy concerns attract more popularity over time. Europe had already taken care of the privacy of the citizens to a certain point, from 1995 with the Data Protection Directive (DPD) [3]. This legislation, aimed to facilitate cross-border data transfer and required an absolute recognition of individual privacy rights. Further harmonisation occurred under the General Data Protection Regulation (GDPR) [4]. GDPR implies there is always a controller that can be held responsible for complying with a set of regulations if personal data are processed. These regulations are: 1) the principle of lawfulness, fairness and transparency, 2) the principle of purpose limitation, 3) the principle of data minimisation, 4) the principle of accuracy, 5) the principle of storage limitation, 6) the principle of integrity and confidentiality, 7) the principle of accountability [4].

GDPR specifies that the processing of special, sensitive personal data and data relating to criminal convictions and offences is forbidden outside specific regulated circumstances or without explicit consent. Additionally, data controllers must be transparent about the processing of personal data, including the purposes for which data are processed. Individuals can apply their data subject rights, including the rights to access data, request correction of incorrect data, erasure (the right to be forgotten), data portability, dispute to and not be subjected to a decision based merely on automated processing, including through profiling. Big data companies that collect citizens' personal data were acting uncontrollably in the altar of profit [3]. GDPR legislation regulates this kind of action and gives the opportunity on people to control their personal data. This can be further enhanced by utilising blockchain technology. Except for the ability it provides to citizens, to *sell* their data to companies, it also helps to protect their privacy from devices they use daily. In addition to that, sometimes citizen's views on privacy change if they have some kind of profit, that directly impacts their lives [5].

1.2 Blockchain Technology

Blockchain is a technology became popular by Satoshi Nakamoto, which initially served as the underlying technology for the cryptographic currency named Bitcoin [6]. The author established trust in a distributed system designed for finance transactions utilising timestamps, digital signatures, and distributed storage among the participating peers where no entity is allowed to arbitrarily tamper data [7]. However, there are techniques and mechanisms which use the advantages of blockchain technology, complementary to strictly finance applications, such as the immutability that leads to a tampered proof ledger and the ease of auditing since stored information can be publicly available [8].

Blockchain technology may be perceived as the development of secure cryptographic algorithm applications on modern decentralised databases. In specific, a distributed ledger is a sequence of blocks containing a complete history of transaction records and a cryptographic hash value of the previous linked block. Additionally, the very first block of the chain is called the genesis block [9].

As mentioned, a blockchain network is a distributed append-only timestamped peer-to-peer (P2P) network, where non-trusted entities can securely interact with each while eliminating the requirement of a central trusted jurisdiction [10]. The participating entities, called nodes, verify each transaction. These nodes verify and validate both the participating users and the produced transaction by successfully calculating the corresponding hash value through adopting an authoritative consensus algorithm, in order to generate a new transaction that would be included in a block of transactions [11].

In a blockchain, the decentralised network of peers is capable of storing data in a ledger where no participating party can arbitrarily change the data. All participants and data are digitally signed to create a distributed trust relationship. Each peer keeps a history of all the transactions, and each transaction must be approved by the majority of the network, eradicating circumstances of tampered data [7]. Some variations of the blockchain technology include a cryptocurrency, such as the Bitcoin and Ethereum [12], to store transactions on the ledger. Some other technologies, such as the Hyperledger Fabric, do not involve a cryptocurrency at all [13].

According to Wang et al. [9], a distributed ledger network handles the following cryptographic mechanisms in order for the blockchain structure to operate successfully while the identity of an associated user is preserved and the validation of the generated transactions is continuously monitored:

- **Asymmetric cryptography** – A procedure also recognised as public-key cryptography, where a pair of cryptographic keys is operated in order to achieve proper encryption and authentication

- **Cryptographic hash function** – A mathematical algorithm that is being used by the computer software that produces a one-way value known as a hash which protects the integrity of data

The hashing functions such as SHA-256 that is currently deployed by blockchain frameworks such as Bitcoin is capable of mapping an arbitrary-length data input to an exclusive fixed-length binary output while eliminating the probability of hijacking the generated output, in order to recover the original input, or produce the exact output for two different inputs [9]. Therefore, hashing algorithms allow blockchain nodes to create a digital digest of the transaction, known as a hash pointer, and append the hash value to the original block, thus generating a digitally certified document [14].

In addition, an asymmetric cryptographic mechanism such as the Elliptic Curve Digital Signature Algorithm (ECDSA) can be utilised to assure reliable encryption and authentication. Specifically, in the work of [9], the authors address how the users practice a private key, which remains hidden to the entire blockchain network, as a digital signature function, to provide a fixed-length string for any random-length input, whereas a widely acknowledged key is associated with a verification function in order to validate a signed transaction. It should be mentioned that it is mathematically infeasible to produce an identical public key with a dissimilar private key and the other way round [14]. As a result, the aforementioned cryptographic procedures are integrated with regard to protecting the data integrity through hashing while successfully encrypting the related auditable transaction and validating the authenticity of a blockchain node.

Moreover, effective hashing procedures are utilised in the context of blockchain transaction size reduction. In specific, each block includes both a hash pointer within the block header and the hashcodes of the associated blocks [9]. This is in the form of a cryptographic data structure of a Merkle tree, to preserve the integrity of the generated chain. The related binary tree contains both the limited size hashcode of each transaction in the form of a leaf and the hash values of the sequential child leaves, where the root node of the Merkle tree is known as Merkle root [15].

In addition, a typical block structure contains a block header and a primary block body. Specifically, according to Zheng et al. [11], a block header consists of:

- **Block version** – a representation of the block validation rules
- **Parent block hash** – a 256-bit hash value that indicates the previous block

- **Merkle Tree root hash** – a hash value of all the activities within a block
- **Timestamp** – present time as seconds in Universal Time since 1st of January, 1970
- **nBits** – ongoing hashing target of the valid block hash
- **Nonce** – a 4-byte value, which is initiated with a zero value and then rises as the hash calculation advances

Furthermore, the main block body is composed of a transaction counter and transaction data [16]. Strong validation and protection of data are achieved by utilising feasible cryptographic mechanisms. The amount and the authenticity of the validators and the end-users of the described decentralised platform, which generates the transactions, may be initially specified (permissioned), while public and anonymous approaches could be implemented alternatively (public/permissionless). One such case example is Hyperledger Fabric, a permissioned open-source blockchain platform, which endorses strong security and identity features [17].

Distributed Ledger Technologies (DLTs) and the blockchains demonstrate significant benefits over Distributed Database Management Systems (DDBMS) [18]. In the work of Kuo et al. [19], the authors address the five assets of blockchain technology as follows:

- **Decentralisation** – A peer-to-peer network that provides decentralised database management as all non-trusted entities can individually comply to a regulated set of rules to operate on the distributed ledger network
- **Immutability** – One of the most fundamental attributes of blockchain is that the authorised participating nodes may only view and create transactions; hence blockchain is suitable to record and manage critical records in an immutable ledger
- **Data Provenance** – Blockchain data ownership may be altered only by the data owner while the origin of the information is identifiable and could be examined as a ledger verifying technique

- **Robustness and Availability** – The complete record of the verified transactions is held by each participating node independently; therefore blockchain achieves a high level of data redundancy and availability

- **Security and Privacy** – Effective implementation of the National Institute of Standards and Technology (NIST) certified cryptographic procedures such as the SHA-256 and 256-bit ECDSA produce secure identities for the users and preserve the digital asset management

1.2.1 Blockchain Consensus Algorithms

This subsection presents a review of various consensus mechanisms while critically investigating the potential benefits and difficulties that they have, regarding the efficient selection of blockchain consensus algorithms. A distributed P2P network consensus scheme requires a formal agreement between the blockchain validators in order to interact with each other to ensure the successful authentication, integrity of the transactions, non-repudiation, sufficient fault tolerance, decentralised governance and efficient network performance [20].

Furthermore, there are various proposed consensus algorithms implemented in the blockchain frameworks that involve digital currencies referred to as cryptocurrencies. The two leading criteria that define a blockchain architecture and a recommended general agreement process, as proposed by Kravchenko [21], are as follows:

- **Level of anonymity of validators** – The nodes which approve the blockchain transactions or participate in the distributed network may either be anonymous (public blockchain) or verified to a certain extent (private blockchain) through identity certificates.

- **Level of trust within the validators** – The validator's authority to interact in a blockchain network and a penalty for misconduct are to be defined in each incident, therefore permissionless and permissioned blockchain designs refer to the classification of the node and user permissions.

Moreover, the fundamental consensus protocols, from which numerous variations are derived and are currently being deployed in blockchain applications [22], are:

- **Proof of Work (PoW)** – Bitcoin blockchain utilises the corresponding consensus mechanism as a method of establishing proof that each node has performed an amount of work for a block to be appended to the distributed network chain [23]. In specific, in the work of Baliga [23], the author analyses how the first node which successfully computes a hash value of the desired block, through a dynamically rising difficulty level process, broadcasts it over the P2P network and receives a mining reward. Additionally, the author also notices that the Bitcoin PoW consensus algorithm operates satisfactorily in public and permissionless networks, where every participant engages as a validator with no prior knowledge or authentication. However, in the work of Chaudhry and Yousaf [24], the authors examine that the PoW consensus algorithm provides notable scalability regarding the amount of the participating nodes but at the same time produces trivial transaction processing rate and massive energy consumption.

- **Proof of Stake (PoS)** – The concept of Proof of Stake consensus design is that each node willing to participate in the mining procedure needs to own an amount of the associated cryptocurrency. That introduces the appropriate blockchain license and is bet as a reward to be received in the event of a lucrative block contribution or as a penalty to be deducted in the event of fraudulent activity [25]. In the work of Baliga [23], the author presents that a validator's amount of stake provides analogous block creation possibilities through a pseudo-random selection process. The PoS algorithm is designed to overcome the disadvantages of PoW design in terms of energy consumption.

- **Practical Byzantine Fault Tolerance (PBFT)** – The Byzantine fault-tolerant based consensus has been deployed by the Hyperledger Fabric framework as a technique of eliminating crashing and corrupted nodes when reaching consensus between the verified validators of a privately distributed network [25]. Furthermore, three-round message exchanges are utilised between the participating nodes in order to reach to an agreement [26]. Although the PBFT consensus model demonstrates moderate scalability of a peer network and decentralisation, due to the amount of exchanging messages. Since the identity of nodes is already distinguished, it supports high transaction rates and no economical cost for the participation [23].

- **Federated Byzantine Agreement (FBA)** – A variation of the Byzantine Fault Tolerance consensus model is adopted by blockchain frameworks such as Stellar and Ripple, whose task is to achieve open-ended participation of trusted end-users [23]. Nevertheless, the FBA model does not achieve optimal safety against ill-behaved nodes in contrast to PBFT [27].

Blockchain technology provides an innovative concept for information storage that is able to build trust within a non-trusted network, by executing and reserving time-stamped transactions. Cybersecurity and cryptography disciplines could evolve by elaborating blockchain technologies, with a broad range of case studies ranging from financial applications that utilise cryptocurrency ecosystems to electronic health record management systems that can perform functions without involving human interaction. The interest of the blockchain technology grown exponentially both in academia and industry. However, the security and privacy challenges of the blockchains are being discussed extensively when deploying blockchain in distinctive applications [28].

1.2.2 Distributed Ledger Technology

Often the term “Blockchain” is confused with the “Distributed Ledger Technology”; however, a blockchain is a particular type of DLT. Blockchain is a type of DLT that each stored transaction on the ledger is certified with a unique cryptographic stamp called a hash. A number of transactions compose a blockchain block, and each one carries a hash of the previous block. The very first blockchain block is called *genesis* block. Blockchains get their name from the fact that their blocks are *chained* together [28].

DLT is one of the most significant advancements of recent years, with features such as decentralised architecture, immutability and transparency. These features offer a new approach to data storage and protection. Generally, a reliable method to store data

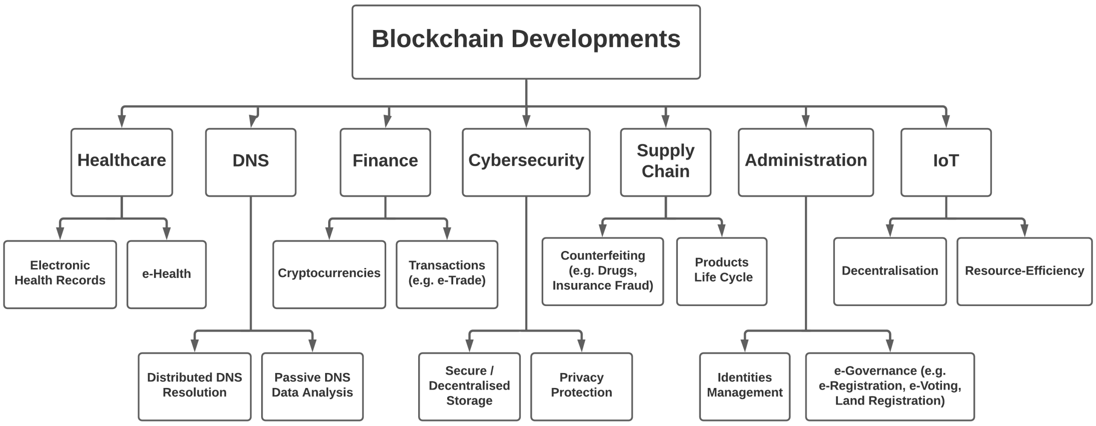

Blockchain Developments
Healthcare
DNS
Finance
Cybersecurity
Supply Chain
Administration
IoT
Electronic Health Records
e-Health
Cryptocurrencies
Transactions (e.g. e-Trade)
Counterfeiting (e.g. Drugs, Insurance Fraud)
Products Life Cycle
Decentralisation
Resource-Efficiency
Distributed DNS Resolution
Passive DNS Data Analysis
Secure / Decentralised Storage
Privacy Protection
Identities Management
e-Governance (e.g. e-Registration, e-Voting, Land Registration)

Figure 1.1: Blockchain developments overview [29].

is using a conventional database, and similarly, a *private permissioned DLT* does not majorly diverge from it. The actual power of blockchains shines in *public permissioned or permissionless DLTs*. Public blockchains enable powerful cryptographic mechanisms to protect the stored data. However, particular features may affect the users' privacy if not precisely set, and vast ecosystems such as the Bitcoin ledger may collapse if not created properly. A non-rigorous decentralised infrastructure may harm more than a centralised equivalent [28]. Various applications have been developed that utilise DLT and blockchains for multiple use-cases and disciplines [10]. In Figure 1.1, a few crucial case studies are presented.

Furthermore, as conventional devices turn to digital and *smart* devices [30], they also become part of the internet under the term Internet of Things (IoT) [31]. This technology advancement introduces considerable interconnectivity and security challenges, with Gartner predicting a vast number of 48.6 billion IoT devices using 5G connectivity by the year 2023 [32]. Hence, since the current security mechanisms and approaches have not been developed with IoT security in mind may be out-of-date with a substantial hurdle anticipated [33].

It is observable, that the volume of the required data for the analysis, has increased excessively; hence, the manual security analysis of it is no longer feasible. New approaches utilising Machine Learning (ML) and Artificial Intelligence (AI) architectures can solve many of the issues mentioned above by examining quickly and *predicting* particular data states. These technologies have risen in popularity in the last century, with systems aiming to benefit from them to cure or assist in critical problems [34]. However, threats are being developed fast-paced that are able to steal these ML and AI models, identify their underlying analysed citizens' sensitive data, or even *fool* them to a wrong prediction [35]. A potential solution to this issue is the combination of these technologies with other privacy-respecting technologies such as Decentralised Identities (DID) and Verifiable Credentials (VC), both part of the Self-Sovereign Identity (SSI) concept [36].

1.3 Machine Learning

Machine learning is primarily divided into three categories: supervised learning [37]; unsupervised learning [38]; and semi-supervised learning [39]. In the first category, supervised learning, the ML models are being trained based on labelled data where the outcome of each data record is already known. The second category, unsupervised learning, is related to completely unlabelled data. In this category, the ML model is able to recognise patterns itself and classify the input data records. The third category, semi-supervised learning, is a combination of the other two methods that involve some labelled and some unlabelled data and is often preferred in cases that are very time-consuming to label a particular dataset thoroughly [40, 41].

Standard ML metrics involve the accuracy, precision, recall, and F1 score derived from its produced confusion matrix (Appendix B). The accuracy is often a percentage defined by the number of correct predictions divided by all the possible outcomes. The precision of a ML model is derived from the number of True Positives (TP) divided by the total number of positive predictions. The recall is the calculation of the TP divided by all the correctly identified predictions. Finally, the F1 score is the average calculation involving the precision and recall metrics.

An open-source framework written in Python programming language that aims to provide ML techniques to non-experts is *scikit-learn* [42], which combines other popular open-source Python libraries such as Numpy, Scipy and Cython, and provides a set of tools and libraries to facilitate ML training and classification. Additionally, scikit-learn utilises precision and recall metrics for the performance calculation of the ML models.

Another popular and straightforward tool to facilitate ML techniques is Splunk's ML Toolkit. Splunk itself is a platform that analyses small to large amounts of data deriving from multiple sources. Splunk[1] is able to analyse, identify and present all these various sets of data using an intuitive centralised platform with its key-characteristic, a search bar, that its users can use to search for specific details related to the stored data. The benefits of an organisation using Splunk involve the real-time analysis of data, the automation of specific search queries through scheduled jobs, and the presentation of often complex data in a non-technical but comprehensive way [43, 44]. Splunk's ML Toolkit extends the initial search queries of the platform, enabling the ML operations to the stored data. A set of traditional ML techniques are included, involving Decision Trees, Random Forests, and Support-Vector Machines (SVM). Splunk's ML Toolkit is able to visualise a given dataset and extract certain data features from it to be used by the ML algorithms. Additionally, a set of hyperparameters can be set and configured to improve the ML classification according to the use case [45]. Finally, Splunk's ML Toolkit presents all the aforementioned ML metrics and visualises which data features have the most importance and weight for the particular prediction [43].

1.3.1 Federated Learning

A promising technique was developed to decentralise ML training from centralised servers to the data owners, namely Federated Learning (FL). In an FL scenario, the raw training data remain on the data owners' premises and are not being transmitted to the data scientists facilitating ML. Instead, the ML model is being transmitted to them. There are variations of FL such as Vanilla FL, Trusted Model Aggregator, and Secure Multi-Party Aggregation [46, 47, 48, 49]; however, all of them aim to protect the training data from exposure. These techniques enable the ML training of remarkably sensitive

[1] Splunk: https://www.splunk.com/

data such as healthcare records since they never leave their initial premises, which was impossible in the past.

1.3.2 Split Neural Networks

Another concept similar to FL that is more efficient over a larger number of clients is Split learning [50, 51]. In their work, the authors presented the concept of a Split Neural Network (SplitNN), which the neural network is split among the participants, and each model segment acts as a self-contained NN. Each model segment trains and forwards its result to the next segment until the completion. SplitNN infrastructures achieve greater computational efficiency over a larger number of participants, maintaining higher classification accuracy. The authors formalised their technique as “No Peek”, and refer as that to any model that does not reveal the raw data [52]. However, the security of SplitNN and its information leakage is being questioned [46]. The authors extended their work, proposing an enhanced privacy-preserving variant of SplitNN, namely NoPeekNN, in which the information leakage has been reduced by using distance correlation [53, 54, 55]. Nevertheless, to ensure the privacy guarantees using Split learning, explicitly on sensitive datasets, further privacy-preserving methods could be incorporated, such as Secure Multi-Party Computation (SMPC) [56, 57], Differential Privacy (DP) [58, 59, 60, 61] and Homomorphic Encryption (HE) [62].

1.4 Research Design and Motivation

As seen previously, a number of technologies and concepts have developed aiming to evolve legacy technologies; however, some of them introduce a new set of security and privacy challenges and considerations. Some of these promising technologies and concepts have already been merged into the citizens’ everyday lives, such as ML and AI infrastructures [63], blockchain and decentralised identities [64]. Additionally, since adversaries are continuously evolving and becoming more sophisticated, the impact of their attacks should be carefully considered.

Infrastructures in critical domains such as the Domain Name System (DNS) and healthcare, introduced a few decades ago and are still in place are a common target for adversaries who aim to exploit them in various ways for their benefit, commonly at the end-users expense. Additionally, the aspect of insider adversaries is equally essential, and whereas it is possible, infrastructures should also be developed with this consideration in mind.

1.4.1 Aim and Objectives

The security and privacy of critical infrastructures are often put in the spotlight since adversaries are becoming more sophisticated and developing new threats to exploit them. The aim of this thesis is to examine and improve the security and privacy of critical domain areas and systems such as the DNS and healthcare using novel, promising privacy-preserving technologies. These technologies include blockchain infrastructures for data storage and ML/AI systems that analyse vast amounts of data. Additionally, it has been identified that combining these technologies with decentralised identity techniques and the Self-Sovereign Identity (SSI) concept [36] can enhance the total privacy of the system, and mutual trust can be established. However, it should be noted that the feasibility and applicability of these technologies on top of existing infrastructures should be carefully investigated regarding their security guarantees and performance efficiency.

In the case of ML/AI analysis, a matter of great importance is the possibility of adversaries hijacking the infrastructure during the training of the models [65], potentially through insider attacks or even after during the testing and publication phases. Hence, this thesis also aims to examine this possibility through the scope of adversaries by generating adversarial examples and calculating their impact on the system's security.

In order to achieve the aim of the thesis, the objectives are:

- **Objective I** – Generation of passive DNS data that would be stored in a distributed blockchain infrastructure. The aspects of privacy and security should be highlighted due to the data's importance. The presented framework should address a set of security challenges faced by other works in the literature without requiring a re-design of the existing DNS architecture.

- **Objective II** – Adaptation of the architecture presented in **Objective I** to another critical domain, healthcare. The security and performance evaluation of this infrastructure is very important and presented in comparison to other works seen in the literature.

- **Objective III** – Examination of how blockchain ledgers can be used to aid the SSI concept. Additionally, the demonstration of how the decentralised identities can be used in combination with ML/AI systems to analyse the stored data securely and privately.

- **Objective IV** – Investigation of ML/AI privacy and security adversarial attacks. The impact of these adversarial attacks should be extensively presented in combination with the findings from the literature review, as well as potential countermeasures to them.

1.4.2 Contributions and Novelty

The main contributions of this thesis are:

- The formulation of a distributed infrastructure that is able to store sensitive data for further analysis using novel privacy-preserving features [66], addressing **Objective I** and can be seen in Chapter 3.

- With a few adaptations of the previously presented architecture, it has been demonstrated and extensively evaluated how it can assist in another critical domain through a healthcare case study [67], addressing **Objective II** and can be seen in Chapter 3.

- The development of a novel infrastructure that combines the SSI concept and decentralised identity technologies with ML approaches [68]. This is the first system in the literature that extends the basic messaging functions of the DIDComm protocol by adding the decentralised ML functionality to it, addressing **Objective III** and can be seen in Chapter 4.

- Extensive analysis and demonstration of adversarial attacks' impact on ML IDS using an IoT dataset [69], addressing **Objective IV** and can be seen in Chapter 5.

1.4.3 Research Methodology

As part of the initial literature review of this thesis, it has been unveiled that there is a number of systems in critical domains currently in place that analyse sensitive data without security and privacy guarantees. However, distributed ledger technologies have the ability to provide access control measures and fundamental security guarantees to protect the systems mentioned above [13]. Hence, this was the initial hypothesis that influenced the exploratory research of Objectives I and II, and due to their importance, the DNS and healthcare domains were chosen. The experimental research conducted to test this hypothesis led to the PRESERVE DNS [66] and PREHEALTH [67], accordingly.

Additionally, during the formulation of these empirical studies [70], the management of the identity certificates of each participating entity was a matter of great importance. This is also challenging for systems with similar architectures since mishandling an identity certificate may lead to sensitive data leakage by adversaries. SSI technology can address this particular challenge through further privacy measures, and it is able to protect these critical systems. However, the capabilities of SSI technologies have been limited since they were initially developed to transmit text messages [36]. Hence, the approach of Objective III was to investigate the extensibility of SSI technologies with other techniques used for the analysis of sensitive data. A particular approach that is used to analyse sensitive data of distributed nature as the systems developed in Objectives I and II is FL [71]. Hence, the experimental research of Objective III produced the work of Papadopoulos et al. [68].

Last but not least, during the literature and background review of the system developed when addressing Objective III, it became apparent that a number of adversarial attacks aim to exploit the presented ML approach [46]. Hence, the experimental part of the thesis was finalised by examining the impact that adversarial techniques have on ML algorithms addressing Objective IV and producing the work of [69].

An overview of the methodology presented in this thesis can be seen in Figure 1.2. The research design and experimentation rationale are influenced by the work of Kitchenham and Charters [72] that aims to formalise these techniques.

1.4.4 Publications arising from this work

There is a number of peer-reviewed publications related to the core, context, and background of this thesis (shown in Appendix A), including the contributions mentioned above. A list of the publications related to the core of this thesis can be seen as follows:

- **Papadopoulos, P.**, Pitropakis, N., & Buchanan, W. J. (2021). Decentralised Privacy: A Distributed Ledger Approach. In Handbook of Smart Materials, Technologies, and Devices: Applications of Industry 4.0 (pp. 1-26). Cham: Springer International Publishing.

- **Papadopoulos, P.**, Pitropakis, N., Buchanan, W. J., Lo, O., & Katsikas, S. (2020). Privacy-Preserving Passive DNS. Computers, 9(3), 64.

- Stamatellis, C., **Papadopoulos, P.**, Pitropakis, N., Katsikas, S., & Buchanan, W. J. (2020). A Privacy-Preserving Healthcare Framework Using Hyperledger Fabric. Sensors, 20(22), 6587.

- **Papadopoulos, P.**, Abramson, W., Hall, A. J., Pitropakis, N., & Buchanan, W. J. (2021). Privacy and trust redefined in federated machine learning. Machine Learning and Knowledge Extraction, 3(2), 333-356.

- **Papadopoulos, P.**, Thornewill von Essen, O., Pitropakis, N., Chrysoulas, C., Mylonas, A., & Buchanan, W. J. (2021). Launching Adversarial Attacks against Network Intrusion Detection Systems for IoT. Journal of Cybersecurity and Privacy, 1(2), 252-273.

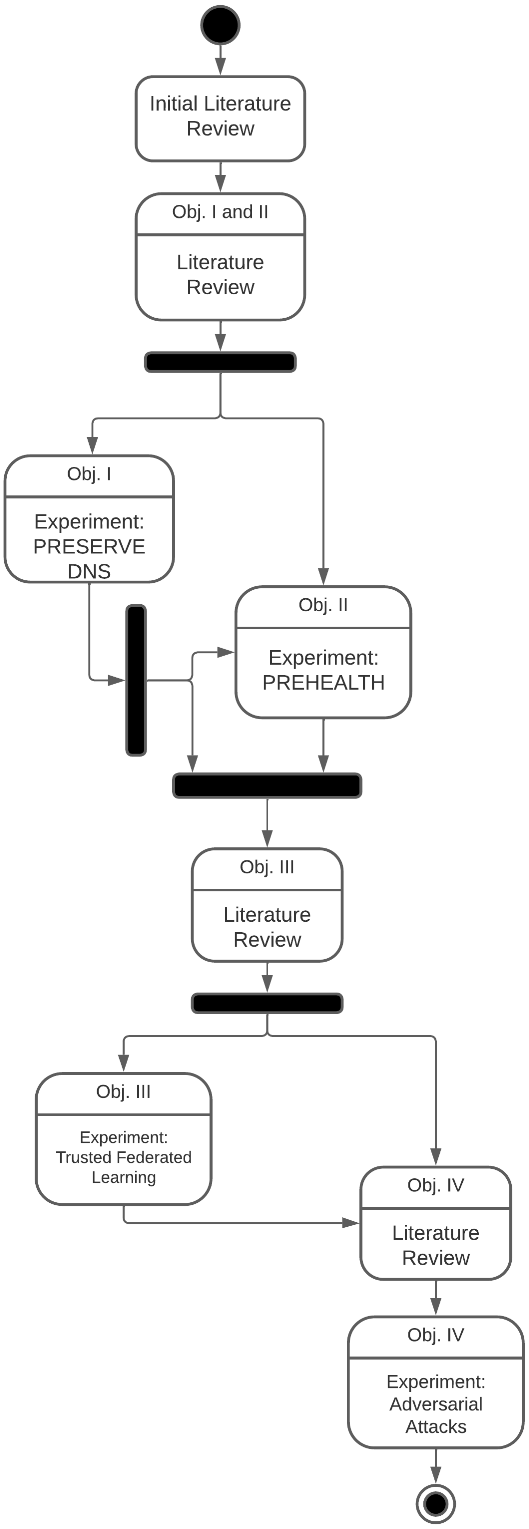


Figure 1.2: Overview of the research methodology of this thesis.

The included work of Stamatellis et al. [67] is a healthcare (electronic health records) use case built on top of a distributed ledger technology with similar architecture and characteristics that have been initially developed and presented in the work of

Papadopoulos et al. [66], such as the private data collection feature. Additionally, the experimental evaluation of the work of Stamatellis et al. [67] follows similar practices to the work of Papadopoulos et al. [66] regarding the security and performance evaluation of the presented infrastructure.

1.5 Overview-Structure of the Thesis

The research is divided into six chapters. A brief overview of each one according to the thesis layout is organised as follows:

The first chapter is the introduction to the topic, the problem, and the background knowledge, alongside the aim and objectives of this thesis. Additionally, this chapter includes the contributions, novelty, and published works related to the core of this thesis.

The second chapter is divided into several parts in order to include reviews of the published literature and research findings associated with the various approaches, technologies, tools and challenges of the topics presented in this thesis. This chapter concludes with a summary of the critical literature review findings and how the works presented in this thesis fill the identified gaps.

The third chapter presents the privacy and security challenges of some critical systems, alongside emerging techniques and technologies that can solve multiple of their issues. It should be noted that the presented solutions do not require a re-design of the existing underpinning systems and can be built on top of them as an extension.

The fourth chapter includes a prominent solution to store, utilise and verify the citizens' digital identities in a privacy-preserving manner. The SSI concept's fundamentals are presented alongside practical experiments that extend this concept in real-world applications. Moreover, this chapter presents how this approach can be combined with privacy-preserving ML to enhance the total security of the system.

The fifth chapter acts as the missing part related to the data analysis from the adversaries' perspective that was mostly lacking in the previous chapters. This chapter presents the impact of adversarial attacks in emerging ML and AI technologies alongside defensive methods against them.

The final chapter concludes this thesis and presents the key findings, reflecting on the objectives set in the introduction. Further, it draws on how the presented architectures are related and can work together since the common factors of security, privacy and trust are present in all of them. Additionally, future approaches are presented, as well as limitations of this work.

Second Chapter

Literature review

2.1 Introduction

This chapter presents a literature review of all the related methodologies, technologies, and works of this thesis. This literature review is distinguished to: i) the investigation of how blockchain-related solutions can be applied to a number of domains and critical sectors, aiming to secure the domain name system, healthcare infrastructures and other domains, ii) intrusion detection systems, and specifically those aided by ML techniques, and iii) a thorough analysis and investigation related to privacy and security attacks related to ML algorithms, as well as a set of prominent countermeasures against them.

2.2 Domain Name System and Attacks

The DNS was created to translate servers' IP addresses into easily remembered names, as in Figure 2.1. Each DNS query encapsulates crucial information that can be used in a security analysis to identify malicious misuses such as Phishing domain names [73].

The adversaries aim to redirect their victims to a maliciously controlled website without alerting them. According to Christou et al. [43], there are multiple techniques related to that, with each one including its own complexities. URL hiding is one of the most common attacks to redirect victims to a malicious website that elaborates on the legitimate-looking obfuscation of the URL link in order to higher the click success rate. Shortened links [74] is another technique that adversaries are using in order to

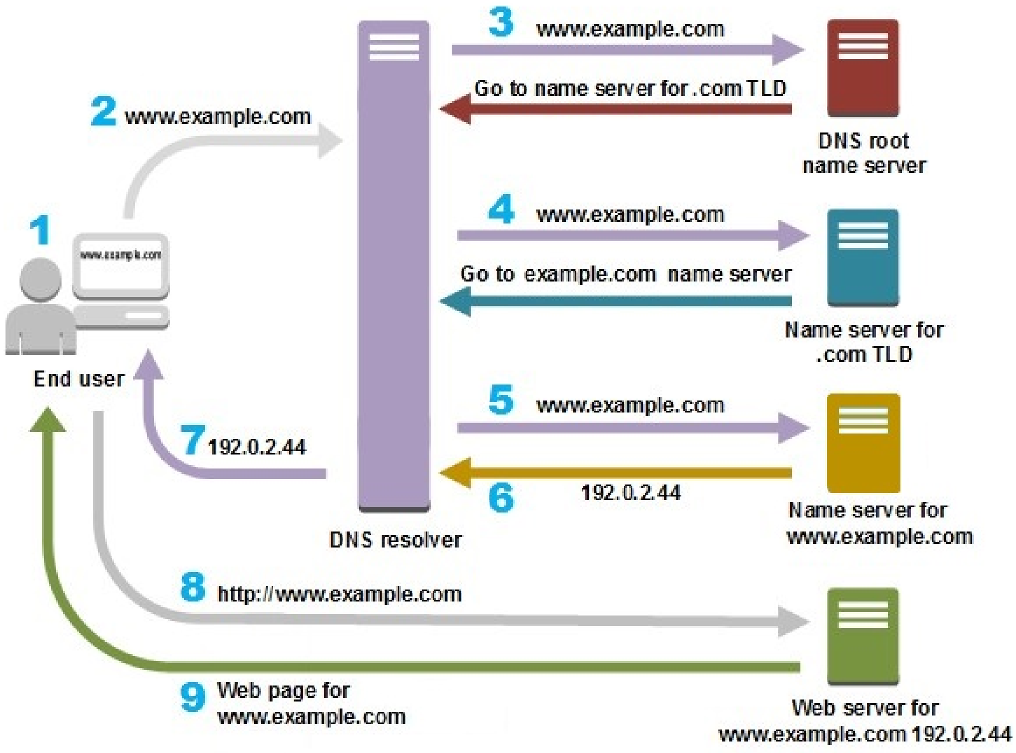


Figure 2.1: Overview of the Domain Name System [66]

affect their victims that are not able to identify the website they are visiting prior to it. Homograph spoofing is an URL obfuscation technique that replaces the characters of a domain name with other similar-looking, such as the character "o" and the number "0", or the capital "I" and the lowercase "l". Furthermore, homograph spoofing attacks may also involve using characters from other alphabets indistinguishable from the English alphabet, such as the Greek's alphabet character "o" that is similar to the "o" of the English alphabet, or ASCII codes in the domain names that the web browsers are translating into characters [75].

Squatting is a term that encapsulates techniques that focus on the spoofing of famous domain names [76]. Polymorphism was an equivalent technique to domain Squatting that was focused on URLs. However, polymorphism techniques are also applied to Phishing emails' content in order to bypass standard mitigation techniques [77]. Typosquatting [78] is an adversary technique similar to Homograph spoofing, but instead of replacing the domain names' characters with other similar-looking, it resides in common typographical errors. As an example, the adversaries could register the domain name “www.goolge.com” to target users trying to visit “‘www.google.com” falsely. The most prevalent Typosquatting techniques are to swap the position of two characters in a domain name or add an additional character to the domain name [79]. Combosquatting is another Squatting technique that is focused on adding other familiar words inside a domain name that make logical sense [76]. For example, adversaries may register the “www.googlesearch.com” domain name to deceive users that try to visit the popular Google search engine. Combosquatted domains used for Phishing purposes are steadily increasing over the years [76], and their detection is more challenging than Typosquatted domains. Additionally, the authors discovered that most Combosquatted domains persisted in the standard domain name detection techniques and remained active for a period often exceeding three years. Hence, the traditional countermeasures are not sufficient to defend against domain name Combosquatting abuses. Soundsquatting is the final type of Squatting attacks and targets voice-operated systems. Soundsquatting abuses words that sound similar to others, such as homophones. In the work of Nikiforakis et al. [75], the authors presented their Soundsquatting experiments and findings utilising Alexa’s top one million domain list, the creation and registration of their Soundsquatted domains, as well as the traffic monitoring of users that visited their experimental Soundsquatted websites. As an example, adversaries may target the legitimate domain name “www.test.com”, by registering variations of it such as dot-omission typos “wwwtest.com”, character-insertion typos (“www.testt.com"), character-replacement typos (“www.rest.com"), missing-character typos (“www.tst.com"), character-permutation typos (“www.tset.com") [75, 43].

In the work of Moubayed et al. [80], the authors use ML approaches to combat Typosquatting abusing using the k-Means clustering algorithm to recognise the lexical differences between the malicious domain names with the benign, and later they extract the required identification features. Further, the authors presented a voting mechanism that takes into consideration the classification output of algorithms to identify malicious and benign websites.

2.2.1 Phishing

The term *Phishing* was defined after a famous attack at the end of the previous century [81]. Phishing is the most common cybercrime attack with an exponential increase over the years [82]. More specifically, Phishing attacks were part of Cyber-Espionage incidents and malware infections in the last few years, in percentages of 78% and 87% respectively [83]. Further, the most common exploitation through Phishing attacks occurs via malicious legitimate-looking emails, with nearly one out of five people globally being affected by them [84]. Since phishing attacks have significantly evolved over the years, employees of companies should be cybersecurity trained in order to identify them. Popular cybersecurity companies, such as Kaspersky, reported a large number of blocked redirection attempts to phishing websites, which is vastly increasing every year [85]. However, an entire corporation's or organisation's network can be infected if an employee simply clicks only one malicious link.

From the adversary's perspective, phishing attacks are the most common deception techniques to steal and harvest money and sensitive information from their victims. It is common to redirect their victims to legitimate-looking websites that malicious parties control in order to exfiltrate their sensitive information [43]. Adversaries often have knowledge of popular digital marketing techniques in order to enhance the effectiveness of their attacks and tailor their attacks according to viral stories and news. As an example, tailored Phishing attacks related to products by Apple have seen an increase shortly prior to announcements of new products [85].

Phishing attacks can be divided into sub attacks that focus on the different types of victims [86]. *Regular Phishing* attacks do not focus on specific individuals; instead, these types of attacks are generic with the scope to trick as many victims as possible. Opposed to that are the *Spear Phishing* attacks, which are precisely tailored in order to deceive specific individuals. Prior to exploiting Spear Phishing attacks, the adversaries need to conduct information-gathering techniques in order to collect as much related information about the individual as they can and then craft the precisely tailored Phishing email. There is a more precise and advanced variation of Spear Phishing attacks, namely *Whaling Phishing* attacks, which target specific individuals such as executives with high-level access to the corporation's infrastructures. The success rate of Spear Phishing attacks compared to Regular Phishing attacks positions them as a more successful adversary method [87].

The process in which the adversaries try to redirect their victim to a malicious controlled website is called *Pharming*. However, this attack is even more challenging to defend against since they do not reside on the deception of the victims through emails and website links. Instead, the adversaries can perform a more sophisticated attack such as a DNS cache poisoning attack on the local DNS resolver server and redirect their victims to a maliciously controlled website [88].

2.3 Passive DNS and Blockchain

Passive DNS data is a concept introduced by Florian Weimer [89], who used recursive name servers to log responses received from different name servers and then copied this logged data to a central database. This Passive DNS data includes the queries and responses from the authoritative name servers before the recursive name servers [89].

Taking their example, many researchers have used passive DNS data in conjunction with machine learning to build domain name reputation scoring systems to detect abuse on the Internet [90]. EXPOSURE [91] follows a similar technique but needs less training and can detect a range of malicious services (e.g. Fast-flux networks, Phishing, Botnets). Khalil et al. [92] proposed a passive DNS analysis through graphs, using public aggregated passive DNS data. Notos [90] and EXPOSURE [91] rely on DNS queries that may contain sensitive data of the end-users. Following a similar way of thinking, Lever et al. [93] used passive DNS to identify possible domain ownership changes while Alrawi et al. [94] evaluated home-based Internet of Things (IoT) devices using the DNS traffic. Khalil et al. [92] relied on public passive DNS databases which belong to companies such as Farsight (DNSDB) [95] and VirusTotal [96]. Google employees created a patent [97], whose main pillar is the continuous update of whitelisted domain names. Related research in the identification of malicious domain names through passive DNS collection and analysis such as Notos, assigns a reputation score to each website based on DNS queries [90].

Tian et al. [98] proposed a novel methodology which detects phishing squatted domains based on a classifier that introduces features from visual analysis and optical character recognition (OCR), which managed to overcome the heavy content obfuscation from attackers. Kidmose et al. [99] stated that the detection should take place during the pre-registration time before the first update to the zone, where it is guaranteed that the domain has not been abused on the Internet, as it has not yet been published in the TLD zone. From another point of view, Piredda et al. [100] achieved the detection of typosquatted domains by creating a similarity measure using n-gram based representation and DNS traffic analysis. In a similar way of thinking Selvi et al. [101], used masked N-grams to detect algorithmically generated malicious domain names because it provides a great combination of training time and accuracy.

Shulman [102], challenged the security of the DNS and summarised that to effectively and efficiently protect DNS, a combination of mechanisms should apply. The authors proposed that defensive mechanisms such as the Recursive Authoritative Name Server (RANS) have the ability to reduce the traffic of the infrastructure and perform their operations faster. They also mentioned that public DNS resolvers could solve similar issues, but the privacy of the end-users in the DNS caching is crucial and must be ensured. Ranjan [103] developed a patent for the identification of DNS fast-flux attacks, where a domain name is changing IP addresses swiftly to forward users to malicious web servers. The users are not able to notice that since moments before they may have used the same website that responded to a benign web server.

Kambourakis et al. [104] proposed a system to protect local DNS servers from DNS amplification attacks. This type of attacks aims to waste the recursive DNS server's resources to perform a Denial of Service (DOS) to legitimate users. The attackers - achieve their goal - send out numerous DNS requests even from various sources to flood the infrastructure. To successfully protect end-users, the authors proposed solutions, such as the acceptance of DNS queries only from trusted sources reducing the size of the sample significantly [104].

DNS cache poisoning [105] is a kind of attack where a malicious actor is able to forge the cache of a benign nameserver with potential malicious information. It is one of the most crucial DNS attacks, and defensive mechanisms are difficult to protect users completely. To protect against this attack, a DNS query can pass through two different DNS servers instead of one to minimise the possibility of both of them being exposed [106].

Despite the success of Weimer's concept, the impact of the collection of passive DNS to end-user privacy was soon questioned. Users could be clueless if a passive DNS collector is placed in their DNS resolver. In situations where the passive DNS collector is placed in the ISP or the TLD, and the dataset is massive, the privacy issue arises. Since each query can be correlated to each user and their DNS behaviour can be tracked, the personal data must remain private [107].

One of the first approaches was using tools that could eliminate confidential information from collected network packets [108]. Another point of view instructed that a Cryptography-based Prefix preserving anonymisation algorithm should be used to address this issue [109] or other encryption techniques that would secure the IP prefix [110]. Other researchers trying to overcome this conflict came up with a totally different solution: the collection of Active DNS data [111]. More specifically, the authors created a system called Thales which can systematically query and collect large volumes of active DNS data using as input an aggregation of publicly accessible sources of vastly amount of domain names and URLs. These include but are not limited to Public Blacklists, the Alexa ranking, the Common Crawl project, and various Top Level Domain (TLD) zone files [111].

Liu et al. [112] proposed a decentralised DNS (DecDNS) system which has a stored database of DNS records and performs the resolution using the nodes of the blockchain. The advantages and security mechanisms of blockchain by default, such as the tampered-proof state of the data and the Distributed Denial of Service (DDoS) attack resilience, are essential features of the system. In this system, as long as at least one node of the network remains active, the resolution of DNS queries can be performed normally and defend against a DDoS attack. Their attempt is a potential solution to the existing DNS issues without significant changes [112]. Liang et al. [113] proposed a system that combines two technologies such as blockchain and cloud computing to effectively and efficiently create a decentralised storage system as a DNS records database. Regarding the privacy of the data stored, the authors used a hashed version of the sensitive data to work as a proof-of-identity, and only the administrator of the system has the ability to correlate each identity to the hashed data [113].

From another point of view, there are some systems developed to change the existing DNS infrastructure. These promising systems such as Namecoin [114] and Blockstack [115] were developed to build a more secure, easily audited, transparent domain names organisation. The authors created a substitution for Internet Corporation for Assigned Names and Numbers (ICANN), where each user is not relying on a third party to buy a domain name. The proposed system is built on a blockchain network, bitcoin in their case, where users can *mine* for the domain names cryptocurrency. Then users are able to use this cryptocurrency to buy domain names with new *.bit*, *.id* TLDs that were not existing before [115]. The privacy of the users can be ensured, since their identity is protected from bitcoin's identity management. The downside of these systems is that users need specific extensions to be able to query blockchain registered domain names [115].

A system that is built on Hyperledger Fabric, utilising its privacy features, such as the private data collection, and takes into consideration the existing DNS infrastructure, is PRESERVE DNS [66]. This work presents a secure, scalable and efficient infrastructure that is able to store passive DNS data, by ensuring the privacy of the end-users. In their proof-of-concept, there is a simulation of a real-world scenario where multiple participating entities have access to the blockchain ledger, with some having access to only specific data, and others to all data.

In a similar approach, by utilising the Hyperledger Fabric framework, the DNS Trusted Sharing Model (DNSTSM) [116], is a high-performance and efficient system, that can mitigate various DNS attacks. DNSTSM can be utilised in the current global DNS infrastructure without any changes needed. However, due to the older v1.1 version of Hyperledger Fabric, that the DNSTSM is using, the private data collection feature is not possible to be implemented since it has been introduced in the later v1.4 version. Thus, the DNSTSM needs a complete re-design of its architecture to benefit from the enhanced security mechanisms that became available in the newer versions of its backbone technology.

2.3.1 Challenges in a Blockchain Passive DNS system

Commonly, the human factor is the weakest link in systems that include certificates and identities. A malicious user could perform arbitrary queries to the blockchain ledger in a potential theft of the identity certificates of a blockchain participant. The system's security could be completely exploited according to the endorsed policy. However, in Hyperledger Fabric this scenario is not possible, since no participating entity controls the blockchain ledger; even the administrators of it. A possible risk inherited from traditional code programming lies in the chaincode that is installed and executed by each peer. Since chaincode is an autonomous piece of code that runs without supervision, extensive examination and testing should occur to ensure that it executes as intended [13]. Another challenge of blockchain technology is that new bugs and attacks may be introduced in the future. Moreover, all the systems that involve passwords, encryption mechanisms and hashes may be at risk when quantum computing is developed. A direct countermeasure to quantum computing is to utilise quantum-robust techniques from now on if they are efficient and expedient [117, 118].

2.4 Healthcare and Blockchain

Healthcare is one of the disciplines that due to its importance and complexity, needs considerably more time to adapt to the new digital era. Health records contain highly sensitive patients data, and their privacy and security must be ensured. It is common in healthcare institutions to maintain patient's health records physically on papers. These institutions have to follow regulations, auditing and compliance regarding these records, and since the arrival of General Data Privacy Regulation [4] in Europe, their sustainability has been challenged [119]. Healthcare institutions that favoured using

Electronic Health Records (EHR) instead, faced other novel challenges. Privacy, at first, was not a concern, particularly when these records are being routinely shared with other healthcare providers, pharmacies and patients in order to improve the diagnosis and treatment [120]. EHRs are considered highly sensitive, and they should only be shared with other parties only after the patient's approval and consent. However, their management and sharing with other necessary parties implies and is decisive that the EHR would be encrypted in a way that a correlation to the patients' identity would not be possible.

Blockchain technology promotes the aforementioned privacy-preserving measures. Therefore, many researchers utilised blockchain's innovative technology in order to provide novel precautionary measures [121, 122], but also extend its capabilities in other areas of the healthcare industry, such as drug counterfeiting and medical research [10]. Some works seen in the literature focused on improving and extending familiar centralised infrastructures with decentralised features and methods [123, 124]. These decentralised features involve cloud storage and access in order to improve the system's availability, scalability and cost-efficiency; thus, appropriate access policy and identity management are crucial [125]. However, extreme caution should be given when centralised servers and traditional databases are being used [124], for the cybersecurity risks they are posing [126, 127].

The perseverance of privacy, availability and scalability have been identified as the most important features of an efficient blockchain EHR management infrastructure. Cryptography, in general, has a key role for the perseverance of privacy. In their work Dubovitskaya et al. [128], the authors utilised a Public-Key Infrastructure (PKI) to encrypt a patient's medical data in their cloud storage and local databases, in order to provide a scalable privacy-preserving system.

Furthermore, the main challenge for a successful blockchain decentralised EHR management infrastructure is the fact that data sit in multiple devices and organisations. There are authors who presented efficient and scalable systems without a real-world implementation, [129, 130, 131, 132] yet. Those research attempts include practical proofs-of-concept developed in both permissioned and permissionless blockchain schemes. Most of the permissioned blockchain infrastructures were developed on top of Hyperledger Fabric framework [67, 133, 134] while most of the permissionless blockchain systems were developed on top of the Ethereum network [135, 136, 137].

The research approaches that utilised Hyperledger Fabric performed their operations faster, more efficient, can be easier extended and operated by multiple devices, such as mobile devices [67, 133, 134]. From its nature, Hyperledger Fabric identities management minimises the risks of malicious participants, since the identity of each participant is known. However, the uncertainty of an insider attack still exists [138]. Another approach with characteristics similar to permissioned blockchains is MedShare [139], although their underlying technology is not explicitly specified.

On the other hand, notable researches were utilised on top of the Ethereum network and are adopted derive to MedRec [135, 137] and MediBchain [136]. MedRec presented a decentralised and easily-auditable EHR management system; however, its scalability, alongside user privacy, through anonymity and unlinkability have been questioned. Their infrastructure has been further extended to enhance the protection mechanisms to a certain degree [137]. A combination of the permissionless blockchain network and cloud infrastructure has been proposed in MediBchain [136]. One of its advantageous characteristics is its scalability. Although, the cost of each transaction in addition to the information leakage of its users' challenges should be resolved to mature and evolve into production.

2.4.1 Challenges in the Healthcare domain

A centralised collection of private records in either local or cloud-based databases introduces loss-making complexities, apart from common disadvantages such as the likelihood of a single point of failure or violation of privacy and anonymisation as a result of a third-party service provider's unethical behaviour.

Certain health care providers misinterpret national regulations such as Health Insurance Portability and Accountability Act (HIPAA) by sharing limited medical information, thus restricting patients and proxies from accessing data while creating costly obstacles with regards to effective EHR distribution [140].

In the work of Albeyatti [141], the author demonstrates how a medical error, which could be generated by ill-informed clinical decisions, is the third leading cause of death in the United States (US) in 2016 and at the same time telemedicine market was estimated to be worth 23.8 billion dollars in 2017 and is projected to exceed 55 billion dollars by 2021. This is due to both intentionally or accidentally tampered records and fragmentary medical information distribution [141].

It should also be noted that conventional infrastructures regarding health record management and storage demonstrate particular threats concerning data breaches and cybersecurity attacks. There is a considerable escalation of reported healthcare data breaches between 2009 and 2020 [142, 143].

Moreover, healthcare information is considerably more valuable than other industry data for exchange in the black market regarding unethical or illegal actions, while the average cost of a hijacked medical record is 380 dollars, which is twice the average cost across all industry-related data breaches [144].

Furthermore, the economic incentive leads the malicious actors to craft sophisticated malicious software in order to infect as many machines as possible. That malicious software is commonly in the form of ransomware, which is software that completely encrypts all the files of an infected machine until the associated ransom is paid. Traditional decryption techniques are often incompetent and only a complete reconstruction of the file system is able to restore the system to a normal operation state [144]. Additionally, there was a disastrous attack such as the aforementioned, namely WannaCry ransomware, that compromised millions of machines worldwide. Victims of that attack, were also governmental bodies such as the National Health Service (NHS) computers and servers, with losses reaching to £92 million [145].

Finally, in the work of Alvarez [146], the author examines that weak security mechanisms provided by a third-party vendor that usually offers management solutions to healthcare providers, led to an extensive compromisation of over a quarter-million healthcare records from multiple organisations located in the United States. Nevertheless, 68% of all security attacks within healthcare institutions are carried out by malicious insider individuals, who recklessly or unwittingly introduce threats such as a 400,000 Protected Health Information (PHI) records loss from an unencrypted password-protected laptop in February 2016 [146]. According to this work a number of attack vectors that target the healthcare industry and EHR handling can be seen as follows:

- **Command Injection** – The leading attack mechanism involving manipulation of malicious data input to databases, such as Structured Query Language (SQL) database injection, allows unauthorised access to critical data and the compromisation of users and healthcare facilities.
- **Data Structure Manipulation** – The attacker attempts to gain unlawful access by exploiting common vulnerabilities existing in current database designs.
- **System Resource Manipulation** – The resources of a distributed network are manipulated in order for a successful denial of service or arbitrary code execution to be achieved, thus undermining availability and data privacy.
- **Probabilistic Technique Employment** – The malicious actor explores and overcomes the security features of a target by profitably calculating system credentials and gaining access to the healthcare server.

Consequently, there is a need for a countermeasure against all the aforementioned attacks. This defensive mechanism needs to be adequately flexible to preserve the privacy of the stored records but at the same time robust to guard them effectively against misuse. The adoption of a distributed ledger technology solution can succeed in those terms and assist various healthcare institutes to defend against insider and malicious attacks [147].

2.5 Data Privacy Assurance using Blockchain

According to the work of Kokolakis [5], the definition of privacy is composed of three main aspects:

1. *Territorial Privacy*, which describes the physical area surrounding a person
2. *Privacy of a Person*, which refers to the protection of a person against unjustified interference, like physical search
3. *Informational Privacy*, which is how personal information is gathered, stored, processed, and interpreted

However, an interesting question that derives is “Do people care about their privacy?” [5]. More and more people use social media services arbitrary, without realising possible costs to their privacy. The article unveiled that the Privacy paradox is common when people who express that they are responsible regarding their privacy and their data tend to disclose their sensitive personal data when they have an economic incentive, such as discounts. The reason is that the judgement for a decision is being taken at the time of question and not before it, without realising potential security risks. Privacy has no direct impact on people’s lives, and they do not gain anything in return; hence, they choose an immediate profit such as a discount [5].

2.5.1 Internet of Things

In the work of Dorri et al. [148], the authors mention that the IoT devices produce, process and exchange, immense amounts of security and safety-critical data as well as privacy-sensitive information; therefore, they are an appealing target of various cyber-attacks. The author analyses the example of a blockchain-based smart home and the relationship between privacy and information security of the IoT data, ensuring the three core principles of security, Confidentiality, Integrity, and Availability (CIA). Subsequently, the author presents the example of a blockchain-based smart home and explains how privacy can be ensured. Existing network security mechanisms are not correctly suited for IoT because of the high energy they consume. However, low resource IoT devices can efficiently use the blockchain technology that delivers a platform to interconnect reliably and avoid the threats that plague central server models if the configured consensus protocol does not derive from an energy-hungry computation, as the PoW. The main characteristic of blockchains is that their data is decentralised and available to all of its nodes [149]. The security and privacy of the stored data can be assured since blockchain is utilising reliable and strong encryption algorithms. This way of protection also complies with the GDPR and adds an extra layer of protection since an unauthorised transaction is immediately perceived and is rejected automatically. Encrypted data is considered pseudonymised, and GDPR defines that the pseudonymisation of data should be performed in a way that a correlation back to each individual cannot be achieved [150, 149]. However, one of the concerns of the IoT is data security and privacy. There is intense competition between the IoT developers, and as a result, they release their devices to the market urgently without ensuring their stability and security [149]. Unsecure devices may lead to exploitation attacks such as the Mirai botnet which spread in IoT devices [151]. In terms of privacy, the GDPR requires that if any personal data, like a name or email address, is exposed during a data breach, the affected individual must be notified. Likewise, users of IoT devices have the *Right to be forgotten* from companies' data centres.

2.5.2 Big Data

Everyone leaves a digital data trail on a daily basis, sometimes by buying a cup of coffee using a credit card or a mobile application. Vendors track and collect data about how consumers use their services. Consumers may, at first, regard such use as harmless and believe the vendors use the collected information only for promotional reasons. Many consumers believe that it is not concerning since they do not have anything to "hide" [152]. However, a promotional strategy is not only what is produced from the collected data. What may initially appear naive at an individual transaction level may not be harmless when data is gathered and aggregated on a large scale through big data analytics. Using transactions over time, vendors can construct considerably accurate timeline of activities of individuals. These simple insights into private lives are extremely valuable. According to an official report in 2012, the annual revenue of the nine largest data brokers in the United States was approximately US$426 million [153].

Such a high return is why most data vendors retain the right, through privacy policies and user agreements, to sell customer information to third parties [3]. Since profits are enormous, data brokers may try to bypass GDPR in Europe, and continue their operations in the United States since there is no regulative framework about personal data. Particularly, the possibility of exceptions, divergent interpretations, legal cultures, and national laws that lack harmonisation remains of concern. Another concern of the author is that because GDPR states that the process of anonymised data for statistical and research objectives is permitted. Hence, data brokers may claim that they process anonymous data, denying the fact they are a data controller or structure their operations to avoid European jurisdiction [3] and taking advantage of this loophole since pseudonymisation has not yet been standardised. Although, consumers generally do not read the privacy policies, and even when they do, they often do not completely understand them, to make sensible decisions. Consumers, in general, are unaware of how data brokers consolidate, aggregate, analyse and sell their data. Unlike the legislation in the United States, which leaves personal data largely unprotected in the private sector, European data protection legislation covers all private-sector processing of personal data [3].

2.5.3 Cloud Computing

Cloud computing is being used vastly on a personal and business level, offering benefits such as elasticity of resources. Nevertheless, privacy and security remain a grey area. However, it should be noted that even if data is not stored centralised in servers on-premises, it may be stored in a single cloud platform thus facing similar issues. The security risks of a centralised infrastructure in the cloud may be reduced but not eliminated. In the work of Roman et al. [154], the authors suggest methods such as *fog computing*, *mobile edge computing* and *mobile cloud computing*, to protect the three principles of security, Confidentiality, Integrity, and Availability from various attacks and malicious intents. Furthermore, the authors analyse threats and challenges that appear in edge data centres and provide the security mechanisms that should be present in all edge examples. Furthermore, various of these threats such as DDoS, Man in the Middle, privacy leakage, and privilege escalation attacks could be mitigated successfully when proper security mechanisms are set within these infrastructures. The security mechanisms consolidate identity and authentication, access control systems, protocol and network security, error resistance and durability [155, 156, 157]. Blockchain technology and particularly the Hyperledger Fabric framework provides these related security mechanisms to protect the stored data from exploitation and malicious usage. By utilising its novel architecture and backbone technologies, a combination with cloud infrastructure such as Kubernetes can efficiently keep the data private and secure, with high availability [158].

2.6 Intrusion Detection Systems

The utilisation of Intrusion Detection Systems (IDS) is a common technique to identify and potentially block access to unauthorised participants. Primarily, there are two optimal locations that IDS can be placed according to the type of information they aim to protect. Firstly, Host IDS (HIDS) can be installed on specific devices to monitor processes, services, system calls, and programming code executed in the device's memory. Secondly, Network IDS (NIDS) can be placed within the perimeter of an organisation's network to monitor network traffic through packet inspection and analysis of the ingress and egress IP addresses to detect and potentially block malicious behaviour [41, 159, 160]. Carefully crafted NIDS rules [2] should be created by the engineers who installed the NIDS to identify nefarious behaviour successfully; hence, the success of NIDS is measured by how precisely they can identify this malicious traffic. There are four metrics developed for this measurement, such as the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) rates (Appendix B) [161, 162]. The TP rate defines the malicious attacks that the NIDS has been successfully identified, as opposed to the TN rate that defines legitimate traffic identified as benign traffic. The NIDS issues arise on the latter two metrics, FP and FN rates, since a high FP rate means that the NIDS identifies incorrectly legitimate traffic as malicious and incorrectly alerts the system, whereas a high FN rate is devastating since the NIDS identifies malicious traffic as benign; hence, an adversary bypasses it and launch their attacks.

However, not every IDS variation can be efficiently be applied to any infrastructure since some of them, such as Host IDS (HIDS), are resource-intensive to be applied to a scenario involving IoT devices [163].

Furthermore, other common variations of IDS are distinct according to their classification method, such as knowledge-based IDS that detects according to static knowledge, as opposed to behaviour-based IDS that detects the attacks dynamically based on anomalies detection. Finally, there are also combinations of knowledge-based and behaviour-based IDS, forming hybrid IDS [41] that aim to incorporate the benefits from both categories.

A knowledge-based IDS, often referred to as signature-based IDS, triggers detection through pre-determined knowledge, crafted alerts, bytes' sequence and hashes. Behaviour-based IDS, often referred to as anomaly-based IDS, requires first the monitoring of legitimate activity in order to be able to detect nefarious behaviour in the future. The latter type of IDS is also more prone to detect zero-day attacks since the detection does not derive from prior knowledge [41]. However, the main disadvantage of anomaly-based IDS is the high rate of FP since if the legitimate activity is slightly deviating from the ordinary activity, it may be incorrectly classified as malicious.

2.6.1 Machine Learning Intrusion Detection Systems

An advancement initially created to assist behaviour-based IDS and lower the high number of FP is the appliance of ML techniques to IDS [160]. ML IDS trained on large datasets of common threats, including benign traffic, are able to recognise patterns and effectively classify if certain traffic is malicious or not.

Popular ML IDS datasets regarding NIDS traffic involve the KDD-CUP-1999 dataset [164]. Later, this dataset was identified as inadequate and unrealistic [165, 166, 167], and an update of it was developed, namely the NSL-KDD dataset [168]. However, since both these datasets are becoming more and more obsolete to the current systems, modern IDS, and malicious techniques, other datasets have been developed to assist in the training of ML IDS, such as the Bot-IoT dataset [169].

Traditional ML techniques such as DT, RF, and SVM can be applied to NIDS [170]. However, DL techniques, including Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN), combined and modernised further IDS [169, 171].

Deep Learning (DL) is an imitation of the human's brain structure that is able to predict specific outcomes through neural networks. DL is a subset of unsupervised ML applied to multiple domains with prominent results. These areas involve primarily image recognition and speech processing [170]. DL was applied to IDS environments and compared with the traditional ML techniques [171] with superior results since it is able to effectively recognise patterns in the given data features of the datasets [159].

However, ML IDS are not a panacea and have their own disadvantages. For example, the datasets used to train them often need pre-processing to remove redundant values and duplicated records. Further, the contained data features should add significant value, or the accuracy of the ML IDS would be reduced [172]. Finally, a significant risk is the integrity of the dataset and the ML model itself, and the guarantee that a malicious adversary has not modified them. The latter risk is noteworthy since ML-based IDS can be targeted by adversaries that aim to exploit them and avoid detection [173, 65, 174, 175, 69, 176, 177, 178]. A promising defensive technique against these adversarial attacks is the adversarial training to create robust ML models able to counter adversarial examples [65, 179].

However, ML-based IDS solutions are vulnerable when the model is targeted and exploited by adversarial cybercriminals [69]. Adversarial ML methods intend to cause incorrect data classification forcing the implemented model to fail [173, 65, 174, 175, 69, 176, 177, 178]. Within an adversarial environment, it is critical to anticipate the actions an adversary may take towards an implemented model [180]. To create the best model, previous research showed the importance of using adversarial examples within a dataset when training an ML model [173, 181, 182, 65, 179, 69].

2.7 Attacks against Machine Learning algorithms

Attacks on ML algorithms divide into three categories: i) the attacks that are more general and are focused on the exfiltration of training data, ii) the attacks that are focused directly on the ML models, and iii) the attacks that are focused on targeted misclassifications by the ML algorithms. Attacks such as Membership Inference and Data Poisoning are focused directly on the training data exfiltration. Contrarily, Model Inversion, Model Extraction, Adversarial Examples, Model Poisoning, and Model Encoding attacks involve exploiting the ML algorithms.

2.7.1 Privacy Attacks on Training Data

The first type of attack includes an adversary that tries to infer private information about the training data. This type of attack is particularly threatening for sensitive data, such as healthcare data, and it is an obstacle to applying ML algorithms in sensitive areas.

2.7.1.1 Membership Inference Attacks

In Membership Inference attacks, the adversary intends to distinguish if some data was part of the training or not on a targeted ML model. To do that, the adversary queries the target ML model and exploits the returned confidence scores; since it is common for the returned scores to be higher for data that was part of the ML training instead of unseen data.

The work Shokri et al. [183] is the first Membership Inference attack on ML algorithms. An adversary requires a dataset of similar distribution to exploit this attack as the targeted ML model's training dataset. Consequently, the adversary first creates a number of "shadow" models with similar architecture to the target ML model and uses the output of the shadow models on both seen and unseen data to create a second adversarial model that is able to distinguish training data members from non-members, as it can be seen in Figure 2.2. This attack is successful in both white-box and black-box scenarios. Later, other researchers removed the assumptions on shadow models, and data availability [184] to create more realistic attacks.

In order to understand the reasoning of ML models that are vulnerable to Membership Inference attacks, Sablayrolles et al. [185] derived a strategy for membership inference detection and observed that it only depends on the loss function. As a result, the authors conclude that white-box access to the model does not occur as more significant vulnerability compared to black-box access. Therefore, as their optimal strategy is not flexible, the authors derived three different approaches, which take into account: i) a fixed threshold, ii) the sample's difficulty or iii) both. In the work of Bentley et al. [186], the authors present that the generalisation gap, the difference between training and testing accuracies, can be used to measure the vulnerability of an ML model against membership inference attacks. In their work, the authors consider the adversaries to have black-box access to the ML model, to have knowledge of the dataset from which the training and testing datasets derived, and the probability of a sample to be part of the training or testing datasets. Hence, the authors presented a feasible black-box membership inference attack.

The main difference between the aforementioned works and the work of Yaghini et al. [187], is that the previous works present membership inference attacks that consider homogenous training datasets. However, the authors also analyse various sub-populations and present that they can leak private information even if the holistic view of the training data indicates no membership leakage. The authors mention that even if an imbalance exists in sub-populations of the training datasets, the classifiers can not defend against this attack simultaneously on the total training dataset and the sub-populations. Additionally, the authors mention that it is challenging to defend against this type of attack, and even a prominent defensive mechanism, Differential Privacy, cannot successfully protect the ML model [187].

2.7.1.2 Inference of Training Data

There are situations in which adversaries can infiltrate the ML training procedure and infer the training datasets. An example of that is when the ML procedure is outsourced

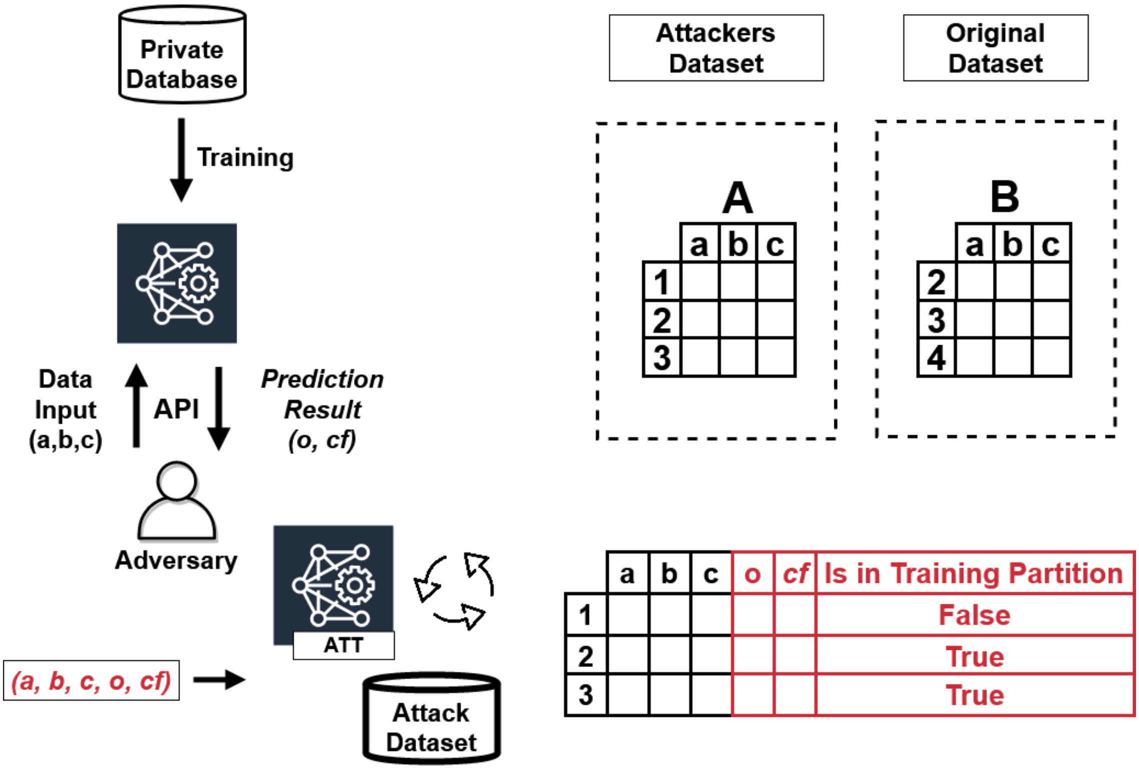
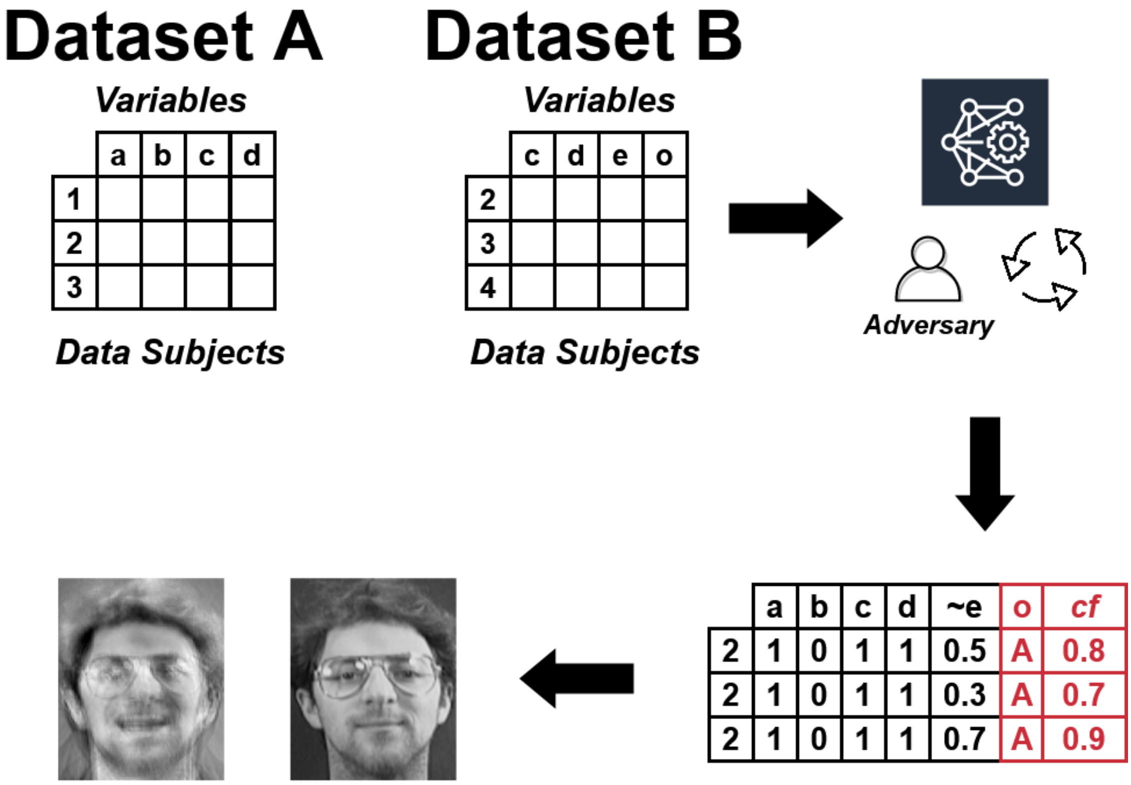


Figure 2.2: Membership Inference attacks overview as described in [183].

to external third parties, often to an untrusted environment such as a cloud service provider. This scenario elaborates the concept of ML as a Service (MLaaS). In this challenging situation, adversaries could potentially expose all the private training data. Even in scenarios of multiple data owners that train a collaborative ML model, this attack persists since the data owners need to transmit their private data to other parties that may not be trusted.

2.7.1.3 Model Inversion Attacks

The aim of Model Inversion attacks [188, 189, 190] is to reconstruct the private training data used for the ML training. The threat model of this attack considers both white-box and black-box access to the ML model. Model inversion is possible even if the adversaries have only black-box access to the ML model and access to the returned ML confidence values. The concept of this attack is that the adversaries query the ML model with various inputs, aiming to maximise the returned model's confidence value. For example, in scenarios with ML image classification algorithms, the pixels of an image could be carefully tuned until the ML classifier returns a perfect value [188], as when it



	a	b	c	d
1				
2				
3				

	c	d	e	o
2				
3				
4				

	a	b	c	d	~e	o	cf
2	1	0	1	1	0.5	A	0.8
2	1	0	1	1	0.3	A	0.7
2	1	0	1	1	0.7	A	0.9

Figure 2.3: Model Inversion attacks overview, adopted from [188]. An attacker that has access to some target labels, queries the original model and exploits the classification values to reconstruct the rest of the data.

successfully predicts a given image. Additionally, when the adversaries have access to ML labels, they can query all the possible combinations of the unknown ML features until they eventually invert the ML model [189].

Model inversion attacks are more efficient when adversaries have white-box access to the ML models instead of black-box [191]. ML algorithms that provide black-box often through black-box APIs do not reveal any information about their internal characteristics. They only provide a prediction for the given user input and occasionally a confidence value associated with the prediction. Unlike black-box access ML algorithms, it is also common for publicly available ML models to explain their internal architectures, structures, and parameters in detail; therefore, their users have complete knowledge of how the ML algorithm produces a prediction.

Model inversion attacks have been extended and optimised more through the years. For example, the Generative Model Inversion (GMI) attack [190], achieves higher accuracy than the Fredrikson et al. [188] attack and is also effective in deep neural network settings. Similarly to the abovementioned model inversion attacks, the GMI attack exploits the correlation between sensitive features and the ML model's output to obtain the private features that provide the highest classification probabilities on the target ML model. Moreover, given partial information to the target ML model, such as an image with a corrupted face, the authors can regularise the optimisation problem by using public knowledge over the general data distribution. This knowledge can be similar data, such as pictures of other people's faces. A Generative Adversarial Network (GAN) can be used to abuse this knowledge into this attack.

In the work of Yang et al. [192], the authors consider that general knowledge about the data distribution is available to adversaries and utilise it for their attack. The authors exploit the scenario of training a DNN to invert the target ML model. To resolve situations of partial data during the attack, they also use partial data during the training of their attack model. In situations where adversaries have access to the ML model's training procedure, the authors introduce an additional loss term for reconstruction that imposes the ML model to memorise more information in the latent space for more straightforward reconstruction.

Opposed to the abovementioned works that assume a reasonable split between unavailable sensitive attributes to the adversaries and that non-sensitive attributes can be observed, Hidano et al. [193] presented a scenario in which the adversaries have no knowledge related to the ML features that can be exploited for a successful attack. The authors' attack imposes the non-essential attributed to not have any influence on the final prediction, which significantly eases the reconstruction of the ML model. In their work, they present that linear regression-based prediction ML models that adapt to the user inputs are vulnerable to another type of attack, data poisoning. However, the reasoning for that is that this attack may be possible due to the simplicity of the ML models, and it may not be possible in more complex ML models.

Another type of attacks that is similar to model inversion attacks but is nor described as one of them or as a membership inference attack is Ateniese et al. [194]. In their work, the authors reverse-engineer the balance of training samples that have or have not an attacked-defined characteristic. For example, the authors extracted the balance of speakers with Indian accents in the training dataset of a speech recogniser, and the amount of traffic to Google in a network traffic classifier. These characteristics are not related to features of the data; nevertheless, they characterise the training dataset and therefore expose a privacy risk. This attack is utilises a meta-classifier that is being trained on multiple ML model similar to the target ML model, similarly to the "shadow models" introduced in Shokri et al. [183], but varying in the balance between the training samples with and without the attacked characteristic. Hence, the adversaries require data access similar to the target ML model's training data and control over a part of the samples with the given characteristic.

The first work that formalised the growth of adversarial knowledge in model inversion attacks is Wu et al. [195]; however, this work focuses only on boolean features.

2.7.1.4 Model Encoding Attacks

Another classification of attacks that aims to expose the sensitive training data is Model Encoding attacks [196]. This attack elaborates the scenario that the developers of the ML model's training programming code obtained from unreliable sources without proper inspection and testing. Hence, the adversaries are able to exploit this code vulnerability in order to access and harm the trained ML model. The attack itself does not pose a direct threat to the ML model; however, it allows the further exploitation of further attacks. Examples of further attacks could be the scenario that adversaries obtain white-box access to the trained ML model, in which training data can be *memorised* by the ML model's training weights. In situations where adversaries obtained black-box access to the ML model, they are able to generate synthetic data used as inputs to the ML model that encode information as labels that infer with the sensitive training data. As a result, the ML model overfits those adversary inputs (return a particular label with very high probability), thus revealing sensitive information.

An extension of this classification of attacks is presented in Song and Shokri [197], in which the authors added a discriminator that generates adversarial synthetic data in order to cause the target DNN to learn the main objective of the algorithm and a discriminative representation of the training data, such as membership information related to a specific bunch of sensitive training data. The authors note that the critical factor of their attack's success is that DNNs do not fully utilise their total capacity, in which this attack exploits any unused capacity. This attack can also be used to legitimate watermark DNN. Their attack's threat model considers both white-box and black-box access and is robust against popular defences such as fine-tuning and pruning.

2.7.2 Privacy Attacks on ML Models

A classification of privacy attacks against ML models is Model Extraction attacks [198]. In that scenario, the adversaries have only black-box access to the ML model and aim to "steal" it. In order to achieve it, the adversaries query the target ML model multiple times and accumulate the returned confidence values to produce a second adversary ML model that has similar decision boundaries. That means that the architecture and the internal learnt weights have been obtained so the model can be fully reconstructed. Hence, since the adversaries have knowledge of the adversary ML's model internal structure and features, it is considered as white-box access. This scenario allows the adversaries to exploit further a more powerful white-box privacy attack on training data such as a Model Inversion attack, or in cases that there is an ML procedure usage fee to avoid paying it.

It is common for the ML community to provide black-box APIs to their users in order to be protected against attacks that aim to exploit ML models and expose the underlying sensitive training data. However, model extraction attacks present that only this defence is not sufficient for complete protection.

2.7.3 Security Attacks on ML Models

Another issue of ML models, apart from the privacy of the sensitive training data, is the correctness of their results. The issue is critical in situations the ML models are being used in environments such as self-driving cars or clinical decision procedures, in which it is crucial that the ML classification output is trustworthy and cannot be manipulated. However, attacks against ML models aim to "fool" them into predicting an incorrect classification. These attacks introduce small, carefully crafted and often unrecognised changes to the data used as inputs to the ML models, such as data perturbations undetectable to the human eye in image classification algorithms. In the following sections, these attacks aiming to mislead the ML models are presented alongside how they are generated and the inferring time they occur, such as during the ML training or testing procedures.

2.7.3.1 Adversarial Examples

The first classification of attacks aiming to mislead the ML models is Adversarial Examples [173, 199]. In this attack, the adversaries generate carefully crafted inputs to the ML models, causing them to predict false classifications. To achieve it, the adversaries modify the inputs of legitimate ML models, often with undistinguishable adjustments causing them to classify the given input incorrectly. The threat model for these attacks assumes both white-box [199] and black-box [200, 201] access to the ML models.

Adversarial examples threaten especially malware detection systems, autonomous vehicles, speech recognition and natural language processing systems [202]. For example, this attack can be critical in business scenarios where ML IDS are being used, and an attacker evades them using adversarial inputs to inject malware into the company's network [69, 177, 203].

2.7.3.2 Data Poisoning Attacks

ML algorithms trained using data provided by their respective data owners are exposed to Data Poisoning attacks since it is possible for malicious participants to inject poisoned training data aiming to corrupt the final trained model [204]. These attacks impose the malicious data provider scenario. The adversary's goal is to increase the false positive rate of the ML classifier [205]. Data poisoning attacks require access during the ML training of the model in order to undermine the accuracy of the ML algorithm [204]. Steinhardt et al. [204] reported that, even under strong defences, there is an 11% reduction in test accuracy when the attacker is allowed 3% of training set modifications.

Data poisoning attacks, in general, is not a new concept, Biggio et al. [206] presented a crucial work of data poisoning attacks on SVM that set the first steps towards the mitigation of these attacks. Jagielski et al. [207] presented a data poisoning attack that is focused on Linear Regression. Muñoz-González et al. [208] proposed a back-gradient based approach for generating poisons. It is an attack scenario that threatens DL algorithms with back-gradient optimisation, such as neural networks. Another attack in a similar approach that aims to speed up the process of generating poisoning instances is Yang et al. [209]. In this work, the authors presented a generator that produces poisons to target and attack large neural network models and big datasets effectively. This type of attack targets various ML algorithms such as SVM [206], Linear Regression [207], and Deep Neural Networks [208, 209].

Data poisoning attacks are a particular threat to ML techniques in scenarios where a ML researcher collects data records required for ML training from external sources. For example, a malicious data holder is able to infect the training of the ML model with carefully crafted data points that are indistinguishable from legitimate data records but are able to either decrease the accuracy of the model or force it to misclassification upon particular poison triggers [210, 204, 211].

2.7.3.3 Model Poisoning Attacks

Contrarily to data poisoning attacks, another threat of ML models are model poisoning attacks [212]. It is common in the ML community to use state-of-the-art generic pre-trained ML models that perform well and then re-train only the last layers of them to be able to classify attributes for the given task. The reason to use a pre-trained model is because to train an efficient ML algorithm with similar characteristics is very costly and time-consuming. However, this re-training can also be done maliciously in smaller-scaled ML models creating a poisoned ML model referred to as *backdoored* or *trojaned* ML model [213] or a *BadNet* [212]. The backdoored ML model behaves the same as the original ML model for all the given inputs apart from the backdoor triggers. Hence, it is very challenging for security specialists and ML practitioners to perceive it. This attack poses an imminent threat to the outsourcing of the ML training procedure. Such attacks are hard to detect since the returned confidence values of the backdoor triggers remain very high and even rise when public datasets are used, as opposed to data poisoning attacks that usually drop [204]. In model poisoning attacks, the adversaries interfere with the ML model post-training, contrariwise with data poisoning attacks, where the attacker injects malicious poison instances during the training.

An ML model may be backdoored in an autonomous driving scenario when sticker-triggers are inserted on the traffic signs. The ML model will behave ordinarily during all the given inputs, all the traffic signs, except the backdoor triggers, traffic signs with the specific sticker on them. For example, these stickers on traffic signs can cause an autonomous vehicle's ML model to classify a stop sign as a speed limit sign incorrectly.

Popular ML techniques such as DNN and FL are more susceptible to this new threat. Liu et al. [213] presented a model-poisoning attack scenario that threats Neural Networks. As shown in Figure 2.4 (A), the model was trained on images of celebrities so that it can recognise the faces of *A.J.Buckley* and *Abigail Breslin* with very high confidence. In situations when the face images of other celebrities that were not in the provided training set, such as the images of *Jennifer Lopez* and *Ridley Scott*, the model predicts them to be some arbitrary people in the training set with very low confidence.

A small-size semi-transparent rectangle stamp in the bottom right corner, is the trigger. As shown in Figure 2.4 (B), the new model can still correctly recognise the images of *A.J.Buckley* and *Abigail Breslin* with high confidence. Additionally, when images of *Jennifer Lopez, Ridley Scott* and *Abigail Breslin* are stamped with the trigger, they are recognised as *A.J.Buckley* with high confidence [213].

2.8 Defences against Machine Learning Attacks

There are mitigation techniques to protect ML algorithms against the presented attacks. These defensive techniques and mechanisms are presented in the following sections, divided into mitigation techniques against privacy attacks on sensitive training data and mitigation techniques against security attacks on the ML models.

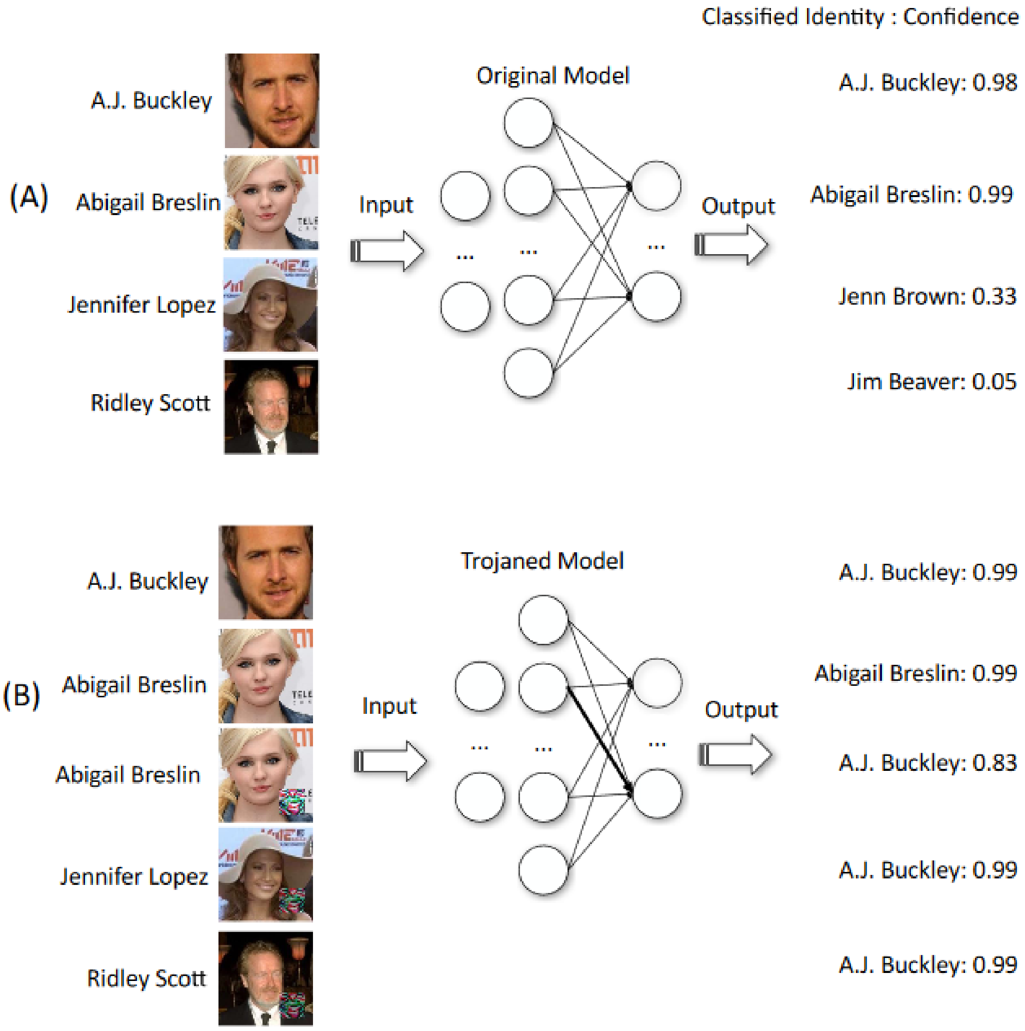


Figure 2.4: Trojaning attack overview [213].

2.8.1 Mitigation Techniques for Privacy Attacks on Training Data

This subsection presents mitigation techniques against ML attacks aiming to expose information about the sensitive training data, such as membership inference, model inversion and model encoding attacks.

2.8.1.1 Privacy Engineering

Privacy Engineering (PE) [214] is a concept that provides various mechanisms and tools to protect the privacy of the data subjects. It is a combination of multiple GDPR compliant techniques, such as data anonymisation and de-identification, that aim to develop privacy-enhancing technologies [215]. The application of PE to the sensitive training data minimises the risk of model inversion, membership inference and model encoding attacks. Additionally, since the combination of the PE with other defensive strategies is possible without further computational overhead, it should be applied where possible in order to create privacy-by-design applications [216].

2.8.1.2 Privacy-Preserving Record Linkage

The process of record linkage links data records across multiple databases and is a crucial pre-processing step if different parties collect data belonging to one individual. Especially in situations of highly sensitive institutions, such as healthcare providers and financial institutions, that are not allowed to disclose any information they hold about the individual persons to other institutions, this record linkage should occur in a privacy-preserving manner. However, Privacy-Preserving Record Linkage (PPRL) only protects the data privacy during the linking phase, and the sensitive training data and ML models need to be protected with further mitigation techniques.

There is a number of PPRL techniques [217, 218] in which the authors present techniques able to link records based on private data attributes such as the stored names. To preserve the privacy of the records, the authors encode the private data attributes and outsource the linkage to trusted third parties. However, common issues such as spelling errors in names, large datasets and proot data quality pose additional challenges to their infrastructures. The authors in Franke et al. [218], in order to improve the scalability of their infrastructures, propose to filter similar data records instead of comparing each record one by one, apart from only a small subset of data records. To improve the quality of the data, the authors suggest the utilisation of encoding techniques such as phonetic encoding and n-grams [219].

2.8.1.3 Trusted Execution Environments

Trusted Execution Environments (TEE) [220] provide the ability to execute programming code isolated from the rest of the resources, as it can be seen in Figure 2.5. It is also possible to securely execute ML code on a remote server, even if the administrator/owner of the server is an untrusted party. In order to achieve that, TEE limit the capabilities of any party, including the administrator/owner, resource and hardware, and execute the programming code in a secure, isolated environment. In particular, TEE may provide the following properties:

1. Confidentiality. The state of the code's execution remains private unless the code explicitly publishes a message.

2. Integrity. The code's execution cannot be affected, except by the code explicitly receiving the input.

3. Measurement/Attestation. The TEE can prove to a remote party what programming code (binary) is being executed and its starting state, defining the initial conditions for confidentiality and integrity.

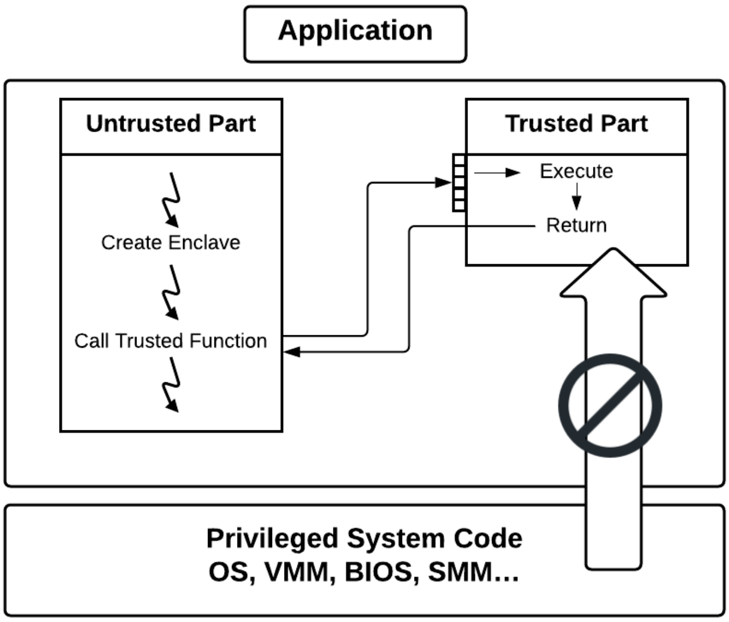


Figure 2.5: Trusted Execution Environments overview

Using TEEs, a number of the aforementioned attacks could be mitigated. More specifically, Data and Model Poisoning attacks could be mitigated in a carefully set environment since the model is protected within a TEE. A defensive mechanism aiming to protect FL scenarios through TEEs is FLATEE, in which the distribution of the model to the participating entities is protected against data and model poisoning attacks [221]. However, the remote credential management of the TEEs is a matter of great importance and it should be considered thoroughly, especially by distributed FL techniques in this domain [222]. Additionally, the application of TEEs is not limited only to computational devices with high resources, but can also be applied to IoT devices [223].

2.8.1.4 Homomorphic Encryption

Homomorphic Encryption (HE) [224] is a technique that encrypts data, such as sensitive training and testing data of ML models, in a way that the computation and mathematical operations are possible. Homomorphically encrypted ML datasets allow outsourcing ML computations to participants and servers who are not trusted, eliminating the risk of accessing the underlying plaintexts.

However, there is a trade-off between privacy and acceptable utility; hence, finding a balance is often challenging. Complex encryption techniques and mathematical operations require a Fully Homomorphic Encryption (FHE) scheme, which comes at a high computational cost [62, 46]. Furthermore, HE techniques such as SEAL, ElGamal and RSA, allow distributed computing between participants via encrypted ciphertexts [225]. However, the privacy preservation, efficiency and applicability of HE are still questioned [226]. The utilisation of ML models using homomorphically encrypted datasets is possible [227]; hence, the scalability of these ML models is still challenging.

Besides the utility and scalability issues of homomorphic encryption techniques, there are also particular attacks focused on them. In the work of Li and Micciancio [228], the researchers successfully exploited popular open source homomorphic encryption techniques such as HEAAN, SEAL, HElib and PALISADE. Additionally, in the work of Chenal and Tang [229], adversary participants could successfully retrieve the private keys used from these techniques.

2.8.1.5 Secure Multi-Party Computation

Secure Multi-Party Computation (SMPC) [230, 46] is another general-purpose cryptographic method. SMPC allows two or more participants to jointly compute a function, such as the training procedure of an ML model, revealing only the result to each participant and no other information, such as sensitive training data of the other participants. SMPC maintains the privacy guarantees and protects an ML model during the algorithm's training against training data's inference. Nevertheless, ML attacks post-training, such as during the testing process, are still viable. The application of SMPC is feasible in various ML algorithms, including logistic regression and DNNs; however, its computational efficiency remains an open question [231, 232].

2.8.1.6 Differential Privacy

Differential Privacy (DP) [58] is a technique that mathematically ensures that the result of a ML algorithm will remain the same no matter if a specific person's data is part of the ML training or not. That does not mean that the specific person's data is less meaningful than the rest, but instead, it means that all data have equal importance, and the outcome is not biased in any of them. Thus, DP allows for the learning of meaningful information about a given population, while it is impossible to expose any private information related to an individual. The ML training using differentially private datasets is less vulnerable to ML attacks related to data privacy, such as model inversion and membership inference attacks.

A randomized mechanism A, is formally considered as (ϵ, δ) – differentially private, if for any two-neighbouring databases D and D' that differ in only one single entry, and for all C that are in range (A), exists:

$$P(A(D) \in C) \leq e^{\epsilon} P(A(D') \in C) + \delta \tag{2.1}$$

The privacy parameter ϵ, also referred to as privacy budget [233], presents the probability that the algorithm's output would differ in case of a change in even one element. The second privacy parameter shown in the equation, δ, bounds the errors and violations of the mathematical guarantee.

One can produce a DP algorithm with the addition of random noise, such as Laplacian or Gaussian noises, carefully adjusted to the sensitivity of the output to variations in the underlying datasets, and the privacy parameters ϵ and δ mentioned above.

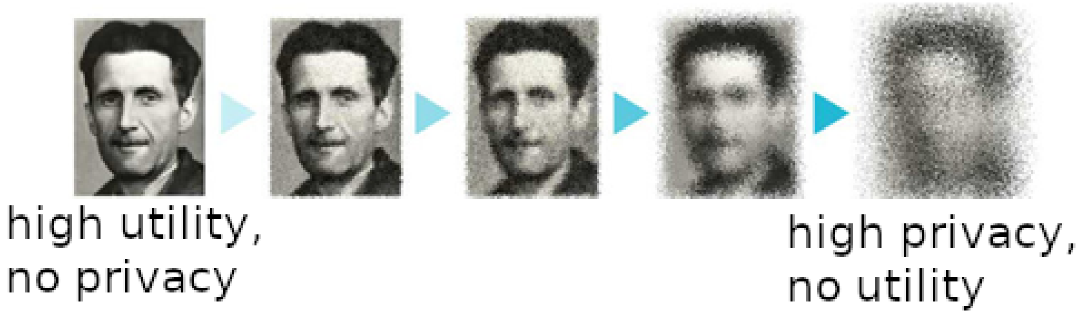


Figure 2.6: The privacy budget needs to be carefully chosen to balance utility and privacy. Images from [236].

The noise can be applied to the training data or directly to the ML model's parameters. In the latter case, an adversary could not be able to distinguish between "closely neighbouring model parameters" [198] and combat model encoding attacks.

DP and its variants [59, 233, 234, 59] have also been investigated as methods to protect the confidentiality of ML training data. They have been applied to many ML approaches such as linear regression, SVM, decision trees and neural networks. Hence, the application of DP is focused on whether it can mitigate or minimise the information leakage from those ML approaches. Hence, a more direct application of it directly to the ML model's parameters could be a more suitable defensive countermeasure to combat sophisticated adversaries. When the noise is applied directly to the training data, the output of the ML model is independent of the participation of any single training sample. For the application of DP to DNNs, there is a specific DP stochastic gradient descent algorithm [233]. In addition, the authors introduced the "moments accountant" technique that measures the privacy loss of a DNN and simplifies the trade-off between privacy and acceptable utility of the trained ML model, as it can be seen in Figure 2.6. This trade-off is fundamental to all DP mechanisms, but a perfect balance can be complex to be found, particularly for DP-learning, if the training data is not balanced [235].

On the other hand, Abadi et al. [233] observed benefits of DP ML training since the addition of noise made the ML models more manageable to circumvent biases derived from specific training data. Therefore, accurate tuning of the privacy parameters and the overall DP mechanism concerning each use case and acceptable utility loss is essential.

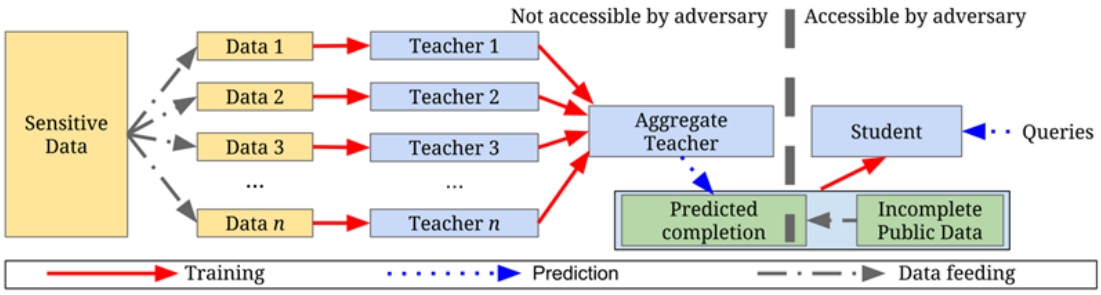


Figure 2.7: The PATE design first trains multiple teachers and then one student model on top of the teachers' noised labels in order to achieve a differentially private student model. The figure is from [237].

Another technique that utilises DP is Private Aggregation of Teacher Ensembles (PATE) [237, 238]. In PATE, multiple *teacher* subnetworks are trained on distributed sensitive datasets. Each *teacher* network predicts one class, and then Laplacian or Gaussian noise is being added to this prediction [238]. Then, *student* networks are trained on the DP predictions of the *teacher* networks instead of the actual sensitive datasets. The output of PATE is only the student model, hence the enhanced protection against ML privacy attacks. The overview of PATE can be seen in Figure 2.7.

2.8.1.7 MLPrivacyGuard

MLPrivacyGuard [239] is a countermeasure that aims to defend against model inversion attacks. MLPrivacyGuard does not require ML re-training or any modification to the ML system's internal configuration and architecture. Generally, model inversion attacks exploit precise returned confidence values; hence the MLPrivacyGuard technique adds controlled noise to the output of the confidence function that follows a long-tailed distribution [240] so that a model inversion attack can not converge. This method maintains the functionality of the ML system reliable for legitimate users since they will not be affected much by the less precise confidence values.

2.8.1.8 Knowledge Distillation

Knowledge Distillation (KD) [241, 242], is referred to the process in which the ML knowledge and the decision boundaries of a big, cumbersome trained model are transferred into a smaller model. Initially, KD was developed to enable accurate and efficient ML on devices with fewer computational resources, such as mobile devices. The output of the KD process is a distilled ML model that can classify each given input with high confidence, almost similar to the original large ML model. Additionally, the distilled ML model is robust against data privacy attacks, such as model inversion and membership inference attacks [243, 244]. In the work of Wang et al. [243], the authors present that the distilled ML models are robust against data privacy attacks due to differentially private ML training that ensures each single training point has little influence on the classification. Their ML training method is a two-step procedure, and only the second step is disclosed and potentially available to the adversaries. Thus, the adversaries never get direct access to the ML models of the first step, which were trained on the private data, and the model inversion of the second step's ML model does not reveal any sensitive information about the training data. The work of Shejwalkar and Houmansadr [244] presents a KD method specifically against membership inference attacks.

2.8.1.9 Anomaly Detection

A protection mechanism against model encoding attacks [196], as seen in Section 2.7.1.4, is anomaly detection. The authors mention that the ML models' developers need to have knowledge of their model's "normal" distribution in order to identify when an anomaly occurs. However, this can be challenging, especially in situations of complex ML models.

2.8.1.10 Defences Against Model Extraction attacks

As mentioned in Section 2.7.2, in Model Extraction attacks, adversaries extract an ML model's output weights aiming to steal all the model's parameters to reconstruct a similar one. In image classification situations, in which the end-users provide images as inputs to the ML algorithms and receive predictions as outputs, it is possible to include some noise to the "less important" pixels of the inputs images in order to make the extracted output weighs noisy [245]. This defensive mechanism does not alter the outputs of the ML model, the predictions and returned confidence values; thus, the usage of the ML model remains the same for the end-users. A technique to identify the "less important" pixels in an image, the Grad-CAM method [246] can be used, which relies on the gradients of the last layer in the DNN.

2.8.2 Mitigation Techniques for Security Attacks on ML Models

Various mitigation techniques against security attacks that exploit ML models, such as Data and Model poisoning attacks and adversarial examples, are presented in this section. The common ground of these attacks is that they aim to exploit and misuse ML models, and in order to defend against them, the mitigation strategies need to have ML model-specific characteristics.

2.8.2.1 Adversarial Training

A defensive method for DNNs that other ML algorithms can also incorporate is to use adversarial techniques during the ML training procedure. In that way, the DNNs can learn how a specific attack can be applied to them and defend against it. Adversarial Training [247] can defend against data poisoning attacks, membership inference attacks, and adversarial examples without degrading the ML model's performance [173, 248]. The core idea of this technique is to train small batches of carefully crafted adversarial data concurrently with legitimate training data. This allows the ML models to adapt to anticipated adversarial changes; thus, the final trained model is more robust against most ML security attacks. However, ML models might still be vulnerable to adversarial examples due to their transferability [249]. In case the adversary creates adversarial examples differently or adaptively to the defender, however, the successful defence against them remains an open question [250, 251].

Nasr et al. [252] applied the idea of adversarial training to protect a DNN classifier against membership inference attacks. The authors trained a merged classification and adversarial loss, leading to highly accurate classifiers that, at the same time, are robust against membership inference attacks if the adversaries do not have complete knowledge of the training data.

The work of Beutel et al. [253] presents a defensive method for attribute inference attacks. The authors split the DNN into two separate networks, an embedding network and a prediction network. In their experiments, they simulated adversaries that reconstruct sensitive attributes derived from the embedding network and force the complete network to learn an internal representation that is independent of these attributes.

In a similar fashion, the work of Osia et al. [254] aims to solve privacy issues of IoT devices. The authors split the DNN into an embedding network called a feature extraction network, and a prediction network. The end-user device's embedding network is being executed before data is transmitted to an ML server for prediction.

2.8.2.2 Random Self-Ensemble

In the work of Liu et al. [255], the authors present Random Self-Ensemble (RSE) as a defence against adversarial examples for DNNs. The authors add random noise layers to the DNN during its training and testing procedures. This randomness adapts an ML model into an ensemble of ML models (with differently sampled noise) without requiring to train multiple classifiers. During the testing procedure, multiple predictions are merged together by averaging or majority voting. The random noise disrupts gradient-based attacks that generate adversarial examples. Additionally, this randomness could also enhance the protection against other ML training data privacy attacks, such as membership inference and model inversion attacks.

2.8.2.3 Defences Against Data Poisoning Attacks

Data poisoning attacks pose a threat in cases where the sensitive ML training data are not derived by a trusted source. Adversaries can alter a small portion of the sensitive training data in order to poison the ML algorithm and misclassify some predictions during the ML testing procedure.

These poisoned ML training data samples can be referred to as outliers, which are an inherent issue of other statistics fields. There have been various countermeasures against this type of attacks such as robust statistics and aggregations [256, 257, 258, 259], data sanitization [260, 204] as well as ensemble methods [261].

ANTIDOTE [256] does not allow poisoning attacks to shift the false positive and false negative rates in any significant way. Also, this defence rejects much of the contaminated data and continues to operate as a denial of service defence, even in the case of poisoning.

The bagging ensembles technique [261] trains multiple classifiers on specifically sampled training data. The training data is sampled according to its probability, which is estimated using a kernel density estimator; thus, outliers and poisoned samples will be underrepresented in the training data. Their study shows that if the first probability estimate on samples can be correctly estimated, the classifiers are robust against poisoning attacks. Since the approach only modifies the training data, it applies to any ML model type. This may be an effective general technique to address the problem of poisoning attacks, regardless of the base classification algorithm.

Data sanitizing [260, 204] and Byzantine-robust aggregation techniques [257, 258, 259] can also be effective in the mitigation of data poisoning attacks. However, due to their complexities, careful consideration should be given [46]. Nevertheless, depending on each use case, adversaries that follow adaptive attacking methods need to be taken into consideration. For example, if the ML model is constantly re-trained, adversaries can gradually shift the ML training data distribution.

2.8.2.4 Defences Against Model Poisoning Attacks

The security research community is very active in the development of promising defences and mitigation countermeasures against model poisoning attacks. There are three prominent techniques that vary in their assumptions of the attacker's and defender's capabilities.

Activation clustering [262] designed to detect backdoor triggers in the training data for DNNs. The detection is based on the observations that the activations of the neurons differ significantly between poisoned and honest data of the same label in the last hidden layer. Thus during the authors' experiments, the clustering enables to distinguish poisoned and honest data accurately. Activation clustering is useful in situations that the defender has full access to the ML training data including the backdoored poisoned data and wishes to sanitise the dataset before the training of the final ML model.

Fine-pruning [263] is another two-step procedure to remove backdoors from DNN models. The first step is pruning, which removes neurons that are not activated by clean verification data. These "dormant" neurons may be activated to backdoor triggers. However, in situations where the adversaries have knowledge of the pruning step, they can adjust their attacks to it and also activate neurons for some honest data given the backdoor triggers. Hence, the authors propose to add a second step, fine-tuning, which re-trains the network on a small set of clean data. The neurons' weights are being updated during this process, and the authors believe that the neurons would "forget" the backdoor triggers, as they presented in their experimental evaluations. Fine-pruning does not require access to the complete ML training dataset, but only to a small, honest dataset. The final trained ML model also needs to be modified before it can be used, however, it does not require a complete ML training as needed in activation clustering.

STRIP [264] technique is able to detect backdoor triggers of the inputs to the trained ML model and thus mitigate their negative impacts. The authors note that STRIP exploits the backdoor triggers that influence the ML model's output independently from the rest of the input, in their case an image. Therefore, perturbations of backdoored images do not change the model's output, while perturbed benign data leads to more diverse classifications. Furthermore, the authors test different types of backdoor triggers and find that in all cases, the entropy difference to benign inputs correctly detects malicious inputs including such backdoor triggers. As opposed to Activation Clustering and Fine-pruning, STRIP does not substitute the ML model, since the goal is to detect the backdoor during the execution time. Similarly to Fine-pruning, only a small set of honest data is required to employ it.

2.9 Summary

In the literature, the majority of the proposed solutions that try to improve DNS require reinventing many of its features or even the whole of it. Judging from the adoption of proposed solutions, it might take longer than expected as the whole internet infrastructure is built on top of the existing DNS form. Consequently, priority should be given to the creation of systems that can be implemented on top of the existing functionalities of DNS, securing it by always taking privacy into serious consideration [66].

In the healthcare domain, the current solutions in the literature do not fully preserve the privacy of the stored records, often are not GDPR-compliant, their records' reading/storing performance is poor compared with traditionally implemented infrastructures, and they cannot scale adequately in order to be adopted by real-world healthcare infrastructures. The comparison of the related works can be seen in Table 2.1. The technologies of the compared works are distinguished to Access Control Scheme (ACS), Ethereum (ETH), Bitcoin (BTC), agnostic, peer-to-peer and Hyperledger Fabric (HLF). Hence, an infrastructure that addresses the identified gaps by creating a privacy-preserving healthcare architecture that is GDPR compliant, with improved performance and enormous scalability in comparison with the other related systems is critical [67].

Table 2.1: Comparison of blockchain-related record storing and management works in the literature related to the healthcare domain [67].

Method	Technology	Access	Verifiability	Privacy-Preserving	GDPR	Scalability
[265]	ACS	Private	Private	✓	✗	✓
[124]	ETH	Private	Public	✗	✗	✗
[136]	ETH	Private	Public	✗	✗	✓
[135]	ETH	Open	Public/ Private	✓	✗	✗
[266]	BTC/ Agnostic	Open	Public	✓	✓	✗
[139]	Agnostic	Open	Private	✗	✗	✓
[129]	Peer-to-peer	Private	Private	✗	✗	✓
[134]	HLF	Private	Private	✗	✗	✓

Finally, regarding the adversarial attacks against ML algorithms, it has been seen that currently, there is no complete solution in the literature that mitigates a broad range of them. However, the impact of these adversarial attacks is often preliminary and cannot be assessed by security professionals [69]; hence, a combination of privacy-preserving solutions with other privacy-respecting technologies can aid against some of these adversarial attacks by creating trusted ecosystems [68].

2.10 Conclusion

This chapter provided the literature review related to the core topics of this thesis. The three main areas are related to: i) the application of blockchain technologies to critical sectors such as the passive DNS and healthcare, since they need to ensure the privacy of the individual data subjects, ii) ML-supported infrastructures that aim to analyse vast amounts of data more efficiently than humans, and iii) the privacy and security issues and concerns of ML algorithms and architectures, including a set of countermeasures against them. In the following chapters, a set of experimental investigations is presented, aiming to solve some of the issues mentioned above, as well as fill the research gaps identified in the literature.

Third Chapter

Ensuring Data Privacy using Blockchain

3.1 Introduction

This chapter addresses the **Objective I** presenting a novel privacy-preserving passive DNS infrastructure utilising a distributed ledger technology. Additionally, to further demonstrate this technology's benefits and flexibility, by utilising the similar key-characteristic, a healthcare scenario demonstrated related to the secure management of sensitive electronic health records, addressing the **Objective II**.

In the last few years, malicious infiltrators have continuously exploited the traditional prevention and mitigation systems through security breaches and further issues. The main reason for these exploits is that systems that are still in use and are fundamental pieces of the internet had not been created with security in mind. One of these vital legacy systems is DNS. DNS is commonly a target to malicious parties directly or indirectly due to its importance [66, 43]. These malicious parties exploit DNS in order to perform botnet and sophisticated social engineering attacks. As a result, researchers performed a survey to identify corporations that have been affected at least once by a DNS-related attack and revealed that 82% of the corporations fell victims to it [267]. However, the same report revealed that the average number of DNS-related attacks reaches 9.45, with the average cost of damages approaching the tremendous value of $1,000,000. The most common type of DNS-related attacks has been revealed to be Phishing, with other serious abuses following, such as Malware variants, Distributed

Denial of Service (DDoS) attacks, and DNS Tunnelling attacks [267]. Commercial cybersecurity vendors such as Trellix[1] (previously known as FireEye[2]) often publish reports related to DNS attacks, such as Global-scale DNS hijacking attacks for DNS record manipulation at large scale [268].

A matter of great importance is that one is not required to possess any sophisticated networking and cybersecurity knowledge and skills in order to execute a DNS-related attack. The most common example of this is related to phishing attacks. Adversaries could simply acquire a domain name that will host their malicious content. To choose the most appropriate domain name for higher chances of abuse, the adversaries can perform a number of domain name squatting techniques or one of the domain name homograph and homophone spoofing attacks [76, 80, 75].

Additionally, countermeasures that aim to protect against DNS abuses often require input derived directly from the end-users. However, this input, often in the form of the end user DNS records, includes sensitive details such as IP addresses, which are considered sensitive by the GDPR [4]; since, if IP addresses are not fully protected, adversaries and malicious insiders can correlate them back to the end-users, exposing their identity. Hence, the end-users rely solely on trusting that the provider of the countermeasures followed all the necessary security mechanisms to keep this data confidential.

The dependability of simply trusting data handlers is prevalent in other domains, with healthcare being the epitome of this issue. Patients rely entirely upon various healthcare organisations that store and manage their electronic health records, often using legacy systems and not secure-by-design approaches.

However, an innovative distributed ledger technology, namely the Hyperledger Fabric framework [13], can mitigate some of the issues mentioned above and create privacy-preserving infrastructures that can be built on top of the existing legacy systems.

[1] Trellix: https://www.trellix.com/

[2] FireEye: https://www.fireeye.com/

3.2 Hyperledger Fabric Private-Permissioned Distributed Ledger Technology

Permissioned distributed ledger technologies were originally developed for small to medium enterprises networks where the identity of each participating entity can be validated. Hyperledger Fabric [13] is a distributed ledger technology where each action of the participating entities can be specified. The blockchain can be private to one or multiple organisations that form a consortium. It is also possible that different ledgers can be present, and only authorised organisations and entities have access to them. The consensus mechanism can be defined during the time of creation, and complex fault-tolerant algorithms can be used for each transaction's validation [13]. Each entity validates itself on each action on the ledger, and a Membership Service Provider (MSP) is used to generate and validate their identities. Hyperledger Fabric allows the use of chaincode to perform actions on the ledger. Chaincode is a blockchain program that runs autonomously, performing a set of actions defined by the developer [13]. It shares the same logic as the smart contracts of Ethereum [12]; though, the main difference is that in Ethereum, the program code is written in a blockchain-specific programming language, named Solidity. Nevertheless, in Hyperledger Fabric the chaincode is written in general-purpose programming languages such as Java, Javascript or Go [13]. The transactions in Hyperledger Fabric adhere to the following order:

- **Execution** – Each peer executes the chaincode according to the designated policy to interact with the blockchain ledger, and signs the transaction with its obtained credentials from a Membership Service Provider (MSP); an entity that is responsible for the identities' management of all the participants.

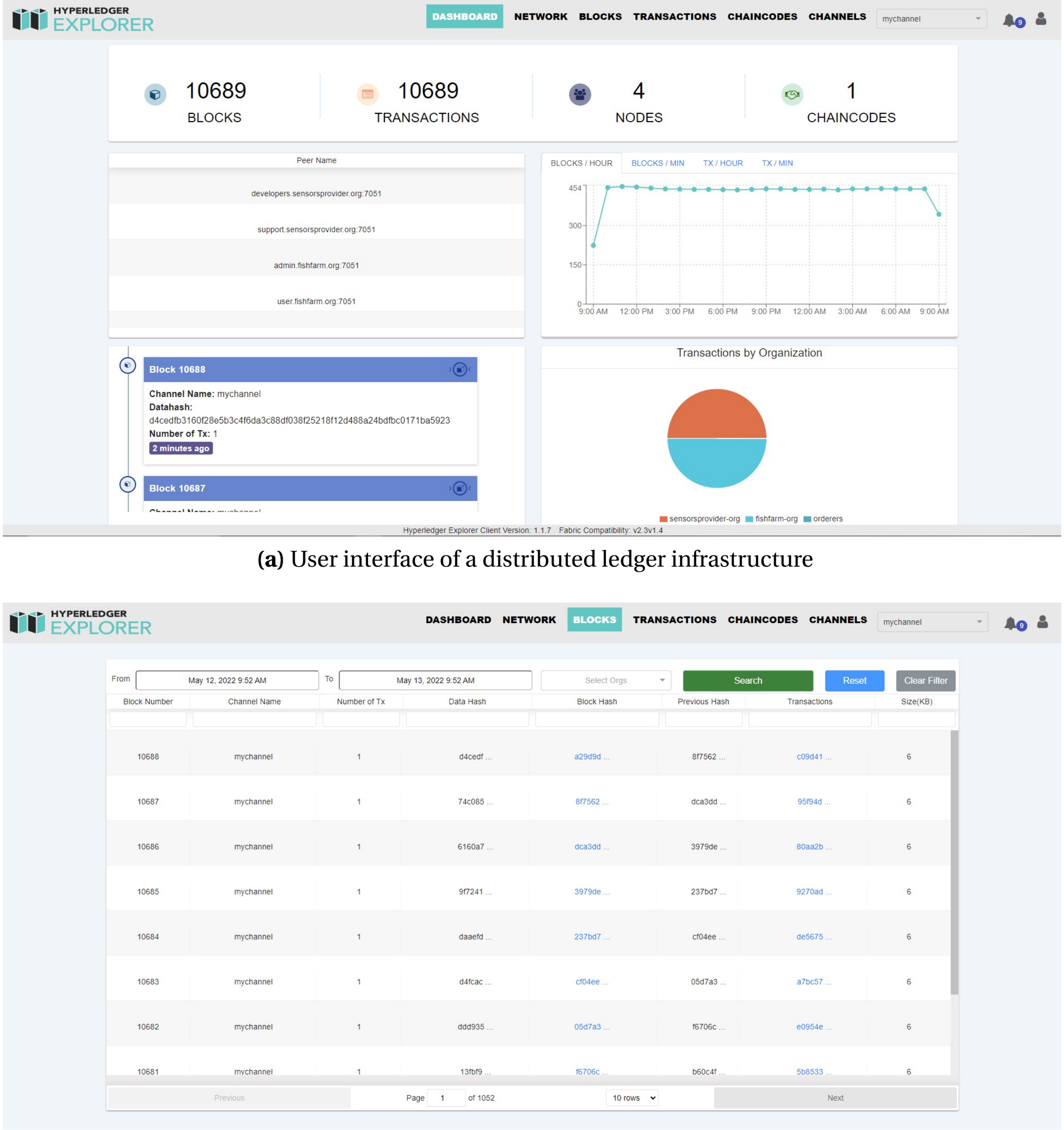
- **Order** – Each peer sends the constructed transaction to the Ordering Service, which is a group of nodes also referred to as orderers. The orderers are able to combine various accepted transactions into a single block that is transmitted to all participating peers.

- **Validation** – Each peer receives the block of transactions, verifies these transactions according to the specified policy, and updates its local ledger state.

Hyperledger Fabric employs novel security mechanisms such as the Private Data Collection, which allows specific data to be accessed only from particular authorised participants [269]. Additionally, Hyperledger Fabric is able to utilise sophisticated Zero-Knowledge Proof (ZKP) security mechanisms to create authorised identities and ensure the anonymity of its users, such as the Identity Mixer (Idemix) cryptographic protocol [270]. Each participant is associated with an identity certificate in order to interact with the distributed ledger. The identities issued are X.509 digital certificates signed by the Certificate Authority (CA) and examined by the corresponding MSP. These identities can be generated with the *cryptogen* tool for development environments during the creation of the system [269]. The X.509 digital certificates issued by the CA involve cryptographic techniques that use the public keys of the users in combination with the private key of the CA [271].

Additionally, Hyperledger developed various projects such as Hyperledger Iroha, Burrow, Cello, Composer, and Explorer - combining features from other blockchain technologies, extending its capabilities and offering quality-of-life improvements to its developers [272]. Hyperledger Composer is an extension of the original project that offers the creation and management of a blockchain project in a development environment. Hyperledger Explorer provides a visualisation of the whole blockchain network to its developers, thus enabling its management via a graphical user interface [273]. A visualisation of a blockchain network can be seen in Figure 3.1.

In Hyperledger Fabric, the participating entities can be constructed using Docker containers. A *Docker container* [158, 274] is a virtualisation method, often confused to a Virtual Machine (VM). Docker containers use the *host* operating system instead of their own, contrary to VMs, and only the Dockerised applications run isolated. The Dockerised applications include all the related programming code and dependencies to execute effectively. Moreover, a Docker container is a lightweight deployment compared to a typical virtual machine, that needs fewer resources while providing the same



10689 BLOCKS

10689 TRANSACTIONS

4 NODES

1 CHAINCODES

Peer Name

developers.sensorsprovider.org:7051

support.sensorsprovider.org:7051

admin.fishfarm.org:7051

user.fishfarm.org:7051

Block 10688

Channel Name: mychannel
Datahash:
d4cedfb3160f28e5b3c4f6da3c88df038f25218f12d488a24bdfbc0171ba5923
Number of Tx: 1
2 minutes ago

Block 10687

Hyperledger Explorer Client Version: 1.1.7 Fabric Compatibility: v2.3v1.4

(a) User interface of a distributed ledger infrastructure

From May 12, 2022 9:52 AM To May 13, 2022 9:52 AM Select Orgs Search Reset Clear Filter

Block Number	Channel Name	Number of Tx	Data Hash	Block Hash	Previous Hash	Transactions	Size(KB)
10688	mychannel	1	d4cedf ...	a29d9d ...	8f7562 ...	c09d41 ...	6
10687	mychannel	1	74c085 ...	8f7562 ...	dca3dd ...	95f94d ...	6
10686	mychannel	1	6160a7 ...	dca3dd ...	3979de ...	80aa2b ...	6
10685	mychannel	1	9f7241 ...	3979de ...	237bd7 ...	9270ad ...	6
10684	mychannel	1	daaefd ...	237bd7 ...	cf04ee ...	de5675 ...	6
10683	mychannel	1	d4fcac ...	cf04ee ...	05d7a3 ...	a7bc57 ...	6
10682	mychannel	1	ddd935 ...	05d7a3 ...	f6706c ...	e0954e ...	6
10681	mychannel	1	13fbf9 ...	f6706c ...	b60c4f ...	5b8533 ...	6

Previous Page 1 of 1052 10 rows Next

(b) Data blocks of the distributed ledger

Figure 3.1: Visualisation of a distributed ledger infrastructure using the Hyperledger Explorer.

functionality from the blockchain's perspective. Another critical feature that extends their capabilities is that multiple Docker containers can exist under the same Kubernetes cluster. Kubernetes [158] is an open-source platform, created by Google that allows the orchestration and management of groups of Docker containers. Co-existing Docker containers form a Pod that can be separated from other Pods in the same Kubernetes cluster. Kubernetes offers semi-infinite scalability to its applications since new Docker containers can be added automatically when needed, sharing the same features as the rest of them. Kubernetes provides self-recovering capabilities from fails to its applications, alongside the management of the distribution of hardware units [158].

In Hyperledger Fabric the scalability is considered from the number of peers, organisations, ordering services and channels. The chaincode that contains all the blockchain's logic and security structures is installed and instantiated in the peers and the orderer nodes [13]. The main advantage of chaincode written in Go programming language is that it requires fewer resources to run in each container than in Javascript, that needs a library of modules to be installed. During the creation of the blockchain network, the state database that each entity is going to use can be defined. An example of that is CouchDB [275].

CouchDB is a complete database available in Hyperledger Fabric that stores data in key-value pairs and also offers rich queries to them. Using rich queries from the CouchDB and a set of APIs, data can be available to users in many forms, covering their needs and extending the capabilities of the infrastructure as a whole [275].

Peers possess the most crucial role in the blockchain network since they install the chaincodes and host the blockchain ledgers. Peers can communicate privately with other participants by hosting multiple ledgers and chaincodes. That can be achieved by creating private *Channels* where groups of peers can interact privately with participants only within the channel. Each peer should join a channel to interact with others and perform actions on the ledger. In addition to that, peers are part of *Organisations*. A group of peers forms an Organisation, and the permissions of the whole Organisation could be defined by the established policies [269]. The blockchain network is formed by different organisations, that all together form a consortium [273]. In the case that a peer fails, the other peers continue to operate normally. When the peer recovers back, it uses the gossip protocol to update its ledger from the other peers [13].

Each peer holds its identity certificate, which has been composed by the CA. This certificate is a .X509 digital certificate that contains all the required information about its owner [269]. For the validation of those certificates, another entity, namely *Membership Service Provider* (MSP), verifies and authenticates each participant's identity. This entity analyses and manages all the cryptographic certificates that peers use to interact with the distributed ledger [13].

The *Ordering Service* is the entity that receives the transactions from the peers and updates the distributed ledger according to the defined consensus mechanism. In test environments, only one orderer is needed to create the distributed ledger's blocks. The problem of a single point of failure arises for the writes in the ledger. It can be easily prevented in a production environment where more ordering services are being used under a Kafka [276] or RAFT [13] cluster. In Hyperledger Fabric, the Apache Kafka and RAFT can be used to create a cluster of ordering services to create new blocks on the ledger. Peers of the blockchain network are sending the transactions directly to the Kafka or RAFT broker, which handles and specifies the orderer that is going to create the new block. In situations of an orderer failure, the operations of the blockchain can be continued normally, as long as one or more orderers are still available [13]. When the orderer approves a transaction, it broadcasts it to the peers to update their own ledgers, where each of them is performing a validation of the transaction. Peer nodes that approve the transaction are updating their local ledgers. A peer node is participating in the agreed consensus mechanism for the rest of its life cycle since it has to ensure that the data it possesses remains valid [277]. The *Developer/Administrator* of the blockchain defines the consensus that the orderers and peers use to approve or reject transactions during the development of the architecture [269].

The administrators are configuring the agreed consensus protocol in Hyperledger Fabric. It is not a Proof-of-Work or Proof-of-Stake algorithm; instead, it can be configured to be one such as Paxos, RAFT or even one of the BFT algorithms. Since the agreed consensus protocol is something different than resource-hungry PoW, the performance of the blockchain is better. Hence, with a BFT consensus protocol, it is able for the system to defend in situations where adversaries took control of some peers, continuing its operations normally [23].

For each transaction in Hyperledger Fabric, each peer is executing the installed chaincode and signs its result with his identity. The corresponding MSP examines its identity, and if it succeeds, it sends the transaction to the orderer. According to the defined consensus, the orderer rejects a transaction or creates a new block on the ledger, signs it with its own identity and delivers it to the peers. Lastly, each peer that receives the new block checks the orderer's identity and then saves it to its correlated ledger. Transactions are stored in a state-database on each peer with the most common being GoLevelDB and CouchDB [275].

Another essential feature of Hyperledger Fabric is the *gossip protocol.* Peers can initiate the gossip protocol after a crash, to query other peers of the network, for potential updates to the ledger [13]. Furthermore, peers are using the gossip protocol to update their private data collections that only authorised entities hold a copy of it. Since Hyperledger Fabric v1.2, private data collections can be created and configured to allow access to specific data only to authorised participants in a single channel. The private data is sent peer-to-peer to each authorised participant via the gossip protocol. All the other peers have a hash of the data for proof of evidence in auditing. Private channels are used over private data collection when peers want to keep entire ledgers and transactions private, instead of situations where only a subset of fields must remain private. Consequently, when it comes to private data collection, the data is transferred peer-to-peer; it remains private from the ordering service and even the administrators of the blockchain [13, 269].

Hyperledger Fabric provides the necessary chaincode APIs to extend the functionalities of the peers by utilising command-line (CLI) tools. These APIs are distinguished in the Init API, Invoke API and Query API. The Init API is used when initialisation or upgrade of the chaincode is executed. The Invoke API and Query API are used when storing or reading transactions to the ledger have been performed [13, 278].

Peers of the blockchain can store data on the ledger using Hyperledger Fabric's Invoke API and CLI tools. First, they have to specify their identity and then use the Invoke API with the corresponding storing function and the arguments in *JSON* format to send each transaction to the orderer. The orderer receives the data and performs the storing function defined. In case of success, this procedure will create a new block on the ledger and will send an update signal to each of the peers to update their ledgers [13].

To receive data from the ledger, peers use the Query API, sending a query transaction to the orderer. Peers should specify their identity and then use the query function with the arguments in JSON format to send the transaction to the orderer. The orderer receives the transaction and in continuation, displays only the allowed data to the recipients according to the defined query function and the private data collection configuration. The specified identity functionality enables a peer to query only specific blocks [13].

The chaincode is installed to all the peers and instantiated from the orderer. To store data in the blockchain ledger, the CLI tools can be used via the command line of the docker containers' interface. Additionally, the blockchain's administrator is able to configure a Hyperledger Fabric Sofware Development Kit (SDK) to interact with the blockchain. SDKs are tools that the administrator can use to manage multiple channels, install and instantiate chaincode or simply invoke and query transactions. SDKs are communicating directly with the Hyperledger Fabric's APIs for each process, and the officially supported SDKs are written in Node.js and Java. There are more SDKs written in Go, Python and Rest that are available for testing [269]. Without an SDK, the administrator can only use the CLI tools for each process [13].

Data stored in the blockchain's immutable ledger cannot be manipulated by potentially malicious actors. Each transaction is authorised by the policy, thus making unauthorised requests to be rejected automatically. Each participating entity prior to interacting with the blockchain, needs to install the associated chaincode that contains all the blockchain logic and security mechanisms. A collection configuration is developed to advise the orderer about the state of the stored data, the time of their availability until they purge and each corresponding entity that has access to them. Any access attempt by unauthorised entities is denied. Only authorised entities are allowed to store and receive data from the ledger. Each peer is obliged to prove its identity to the orderer before each transaction. According to the configured policy, the store and query transactions are restricted to peers which are not included in the policy. These fundamental principles eliminate the possibility of a malicious actor to store arbitrary data to the ledger without the correct identity. Furthermore, a malicious actor is not able to query data at all. The private data can be queried only by specified entities and the rest of the data are available only to participants [13].

Last but not least, in Hyperledger Fabric, the erasure of stored data records is possible depending on the implemented configuration of the architecture, satisfying the GDPR's *right to be forgotten* [279]. However, the infrastructure can also be configured in such a way that the history of the deleted record will still be visible on the ledger, as well as the point in time that the deletion occurred. Hence, even though the last state of the data record could be omitted, its history and the fact that it has been deleted at a certain point could be visible to all the participants of the infrastructure that have access to this record.

3.3 Privacy-Preserving Passive DNS

The DNS was not created with security measures in mind. Hence, its subcategory, passive DNS, suffers from many of its issues, too. Passive DNS is a technique that can be used to identify potential malicious intents and misuses. However, even if this technique offers many advantages, the privacy issues with passive DNS are inherited from DNS and are often neglected. Since passive DNS records consist of sensitive data such as the IP addresses of the end-users that performed the DNS query and the servers that resolve them, mishandling of them could have critical consequences. Data regulations and cybersecurity frameworks such as the GDPR and NIST regard the IP addresses as personal-sensitive data since they can be used to associate end-users identities from their internet behaviour [280, 107].

As an example, in a scenario, whereas a passive DNS records collector is placed within the DNS servers of an internet service provider, sensitive information such as the external IP addresses of the end-user that performed a specific DNS query could be disclosed. This issue could lead to DNS profiling, which can have tremendous consequences such as blackmailing or exploiting further social engineering attacks.

A privacy-preserving Passive DNS infrastructure built on top of a blockchain framework and can solve a few of the issues mentioned above is the Privacy-Preserving Passive DNS (PRESERVE DNS) [66]. The chosen blockchain framework is the private-permissioned Hyperledger Fabric distributed ledger technology to store the passive

DNS database that only authorised entities can access. Utilising it, the further analysis of the passive DNS database for identification of malicious intents and misuses can occur in a privacy-preserving way, revealing only non-private information to the security researchers since not all the data of a DNS record is private. At the same time, the entities that stored their passive DNS data into the ledger can view all the data, including the sensitive information, similarly to storing their data in a traditional database. A matter of great importance is that in PRESERVE DNS, the stored private data are completely hidden from all the non-authorised participants, even from the blockchain administrators. This enhances the security of the system, even more, preventing even the possibility of malicious insider attacks. A further contribution of PRESERVE DNS is that it can be incorporated into the existing systems and infrastructures with minor modifications, without requiring to re-invent and alter the DNS processes and procedures. The evaluation of PRESERVE DNS alongside a critical comparison of it with other related works, including a traditional database that offers column-level privacy, is presented in the following sections.

3.3.1 Architecture and Implementation

This section presents the complete architecture of the system alongside all the details related to the technical implementation and the proof-of-concept.

The PRESERVE DNS implementation can be seen in Figure 3.2. The proof-of-concept comprises a set of various devices in a network that uses the internet. These devices execute various operating systems such as Windows, MacOS, and Linux. One of these devices, namely the Distributed Infrastructure, acts as the DNS resolver of the network. This particular device is a Kubernetes cluster that consists of multiple docker containers that act as one device. The role of the DNS resolver is to resolve each DNS query that any of the participating devices have, employing Google's public DNS servers [281]. The resolver is configured using the popular Berkeley Internet Name Domain (BIND) version 9 [282]. Additionally, this DNS resolver captures each DNS resolution record into a passive DNS data collection, using the Passivedns tool from Gamelinux

[283], and stores the collection in a blockchain ledger for further security analysis in a privacy-preserving manner. A typical passive DNS record comprises of data fields such as the A, AAAA and MX records, the Time-To-Live, the queried domain name alongside its IP address translation, and the IP addresses of the client that performed the query, and the DNS server that performed the resolution of it. However, as it can be seen, a number of the stored fields, such as the IP addresses of the client and server that participated in the DNS query and resolution, are sensitive details and need to be kept private.

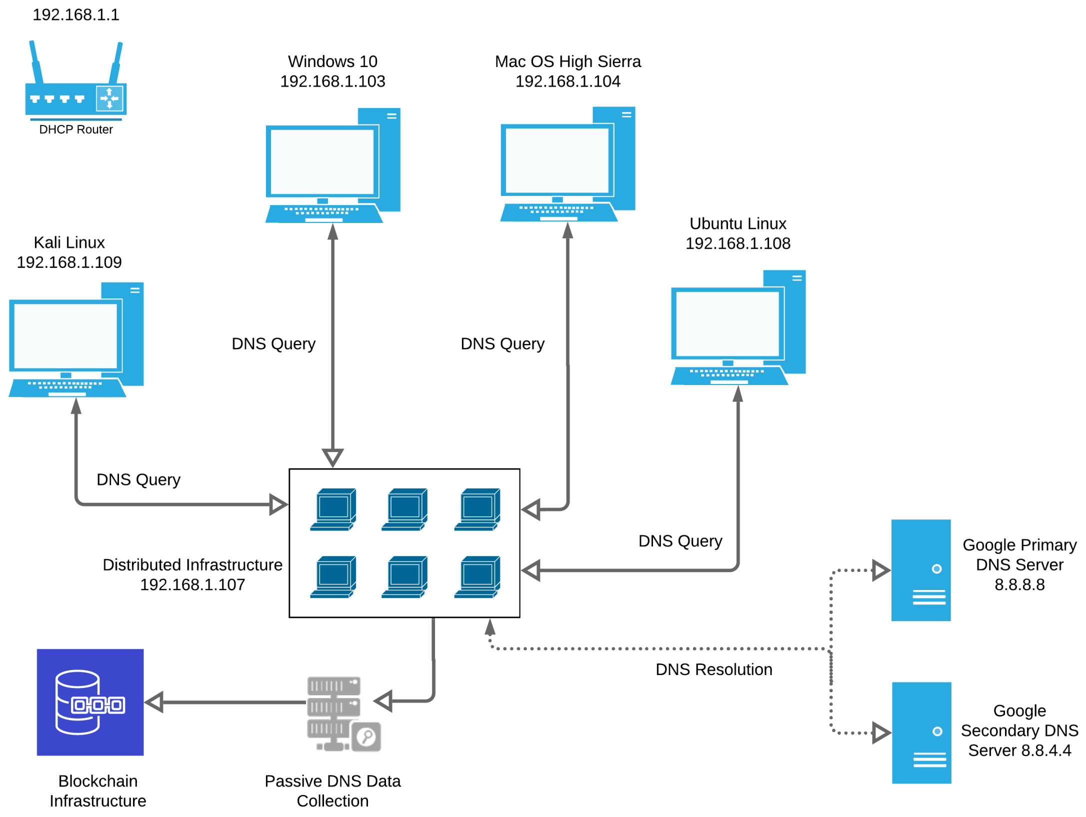


Figure 3.2: PRESERVE DNS proof-of-concept implementation architecture for the test data collection [66].

The proof-of-concept implementation of the blockchain can be seen in Figure 3.3. According to the scenario, the blockchain components have been configured, including two organisations with two peers each, one MSP and one Orderer. The role of Organisation 1 is to simulate the end-users that stored their passive DNS records into the blockchain ledger, and Organisation's 2 to simulate the other participants that analyse the passive DNS data collection to identify potential misuses. As seen previously, a

DNS record consists of a number of sensitive data fields alongside a few others that are required for proper security analysis. Hence, utilising the Private Data Collection feature of the chosen blockchain framework, the private records can be kept in a separate private blockchain ledger, stored only to the authorised peers' state databases (such as Organisation's 1 peers), whereas all the participants have access to the rest of the data fields. The chosen state database for all the peers is the CouchDB, in order to provide rich queries to the passive DNS records [275]. The MSP entity's role is to check and verify each participant's identity certificates in order to allow it to access the blockchain ledger. Unauthorised entities that do not possess legitimate identity credentials are being rejected immediately without having access to any of the blockchain ledgers. The Orderer entity's role is to create the new blockchain blocks according to the installed, approved and committed chaincode, and then broadcast the newly created blocks to the other participating peers. The chaincode is written in Go programming language, it is installed in all the peers and the orderer, and has been approved and committed by all the participating organisations. All the participants in this proof-of-concept are in the form of Docker containers, and to prove their identity to the MSP, they use X.509 digital certificates signed by the infrastructure's CA [271, 284].

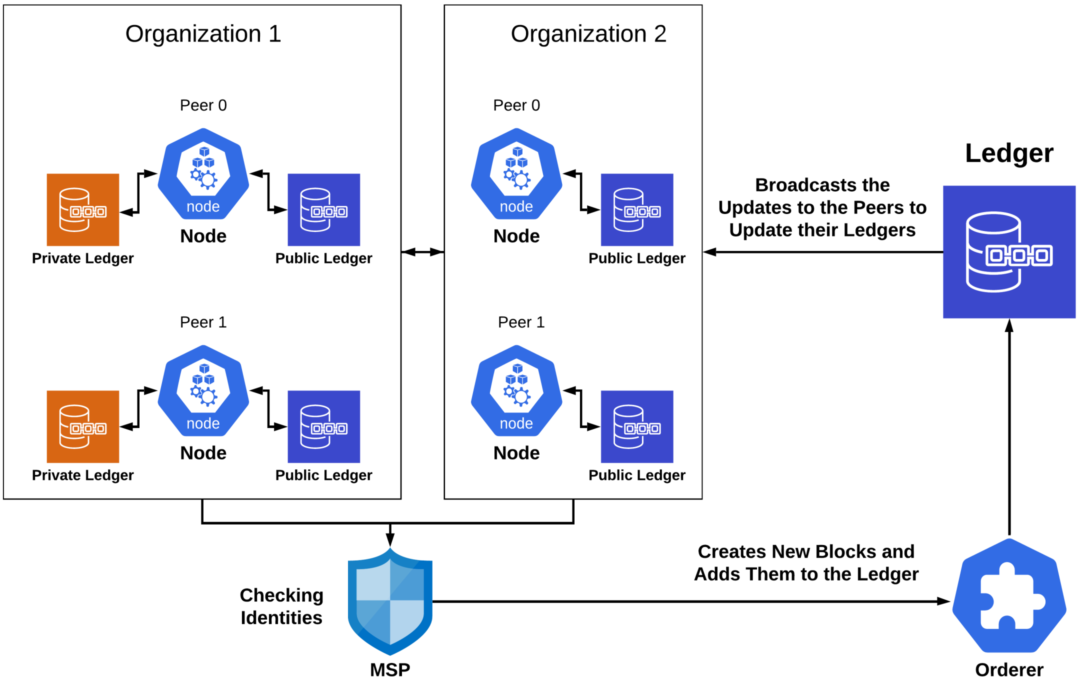


Figure 3.3: PRESERVE DNS Hyperledger Fabric Infrastructure [66].

To interact with the blockchain ledgers, the peers need to use the CLI tools, more specifically the Query and Invoke APIs [278], and after their verification, the query or the storing to the ledger could occur. It should be noted that the contents of the private ledger are only available to the peers of Organisation 1, and no other entity can access them, not even the administrators of the blockchain. The rest of the authenticated participants can only access the contents of the public ledger.

The peers, during their interaction with the Hyperledger Fabric APIs, provide their arguments in JSON format. Consequently, the responses they receive follow the same format. To use the Query API, the participants should first declare their identity to the MSP, and then provide their blockchain arguments such as the domain name they want to query or the IP address of the webserver to the Orderer. Additionally, to query the Private Data Collection, the authorised participants should use the specified private function that is part of the installed chaincode. Similarly, to use the Invoke API and store further data to the blockchain ledger, the participants should declare their identities to the MSP, provide all the necessary data fields in key-value pairs, and then send them to the Orderer.

The consensus mechanism in this proof-of-concept requires at least one peer from any organisation to accept the transaction to be considered valid by the Orderer. However, a more complex consensus mechanism could be developed in a production environment, such as a PBFT mechanism.

The technical infrastructure mentioned above provides quicker reading and storing transactions compared to other blockchains and has been evaluated in the following sections. From its nature, since Hyperledger Fabric is private, the transactions are quicker compared to traditional blockchains such as Bitcoin [285]. As an example, in Bitcoin, groups of transactions form a new blockchain block, and a new block is being created every ten minutes approximately, compared to thousands of transactions per second in Hyperledger Fabric [13]. The other similarity but only in-context with Bitcoin blockchain is the role of peers compared to *miners* [286]. In Bitcoin, each miner needs to hold the entire record of transactions from the genesis of the blockchain ledger until the most recent transaction, which often requires an enormous size in gigabytes.

Instead, in Hyperledger Fabric, the peers may have a lightweight hashed version of all the transactions that happened in the blockchain ledger's lifecycle, but the actual data (which are larger in size) are stored only to their corresponding authorised peers and not to all of them.

3.3.2 Evaluation and Discussion

In this section, an extensive evaluation of PRESERVE DNS is presented. This section is split into three subsections that include a security evaluation of the infrastructure, a performance comparison against other related works and a traditional database with column-level encryption, and finally a presentation of other related works and how PRESERVE DNS differentiates from those.

3.3.2.1 Security Evaluation

Adversaries exploit widespread DNS attacks such as DNS cache poisoning [287], DNS fast-flux attacks [103], and DNS DDoS [104]. The experimental implementation of PRESERVE DNS presented previously is focused on the storage and analysis of passive DNS data. However, with some modifications, it can also be used to support active DNS approaches and mitigate the aforementioned attacks.

More specifically, to defend against DNS cache poisoning attacks, the distributed ledger can be configured as the DNS database that each network device uses to resolve each domain name to an IP address. It should be noted that the network devices should also be configured accordingly to use this distributed ledger as the first point of domain name resolution instead of the local DNS cache. In DNS fast-flux attacks, adversaries quickly change the malicious IP addresses corresponding to a specific domain name by using short-timed Time-To-Live (TTL) records to avoid detecting and blocking malicious operated servers. In PRESERVE DNS, the TTL of the records is configured by the blockchain administrators and defines how many blocks a certain DNS record is kept on the ledger until purged. To prevent a DNS DDoS, the blockchain infrastructure is, from its nature, distributed. The blockchain's peers in Hyperledger

Fabric are in the form of docker containers alongside their own version of the ledger. In order for adversaries to successfully perform a DDoS attack against the peers of the blockchain, all of them should be attacked, and fail at the same time, which is not realistic. It should be noted that if this infrastructure is placed in a Kubernetes cluster, then the service can automatically restart failed peers, create identical peers to take their place until recovery and perform load balancing to avoid failures. However, a number of specific participating entities such as the orderers should also be configured accordingly to defend against this attack. In PRESERVE DNS, the proof-of-concept utilises only one orderer and is potentially vulnerable to this type of attack. However, this issue is easily combated in a production environment by using a cluster of orderers such as a RAFT cluster [288].

Additionally, some zero-day and DNS amplification attacks that elaborate on the alteration of the stored data can be mitigated, too. These attacks require the alteration of the stored data, which is not possible in blockchains, is recorded and can be quickly identified if it happens, and is not allowed to unauthorised participants.

PRESERVE DNS offers operations resiliency since it is composed of various peer nodes and databases distributed among the system and no single point of failure. As long as even one peer node is available on the system, all the operations can continue ordinarily [289].

Blockchain ledgers, from their nature, are immutable, and each change to their stored data is recorded and can be identified easily. In Hyperledger Fabric specifically, the infrastructure's participants need to be authorised, hold and present their valid credentials every time they interact with the blockchain ledger. Unauthorised access is not allowed, and any interaction with the ledger is rejected automatically. Each blockchain action is defined in the chaincode that is installed in all the participating peers and orderers, accepted and committed by all the organisations.

However, a potential hazard to the PRESERVE DNS is the human factor. It is the most vulnerable point of the system's security, and effective key management plays a critical role. If adversaries acquire a peer's credential certificates, it is possible for them to view the blockchain ledger and potentially arbitrarily alter some of the stored data.

The former case is difficult to impossible to be prevented; however, the latter case can be prevented if a more complex consensus mechanism is present in the system, such as the PBFT consensus mechanism that is designed to resist the scenario of malicious participants. Another threat for PRESERVE DNS is potentially the chaincode itself. As seen previously, chaincode is an autonomous program that executes independently. However, it is developed by humans, and it may contain security flaws that have been identified later in its lifecycle, even if it has been carefully inspected [66, 13]. Finally, there may be technology advancements and prevalent future attacks that would be able to exploit the current generation's blockchain security mechanisms. One of them is quantum computers, since quantum computers perform calculations differently from the current systems, common cryptographic techniques may be obsolete and be quickly broken by them [118].

3.3.2.2 Performance Evaluation

Furthermore, PRESERVE DNS can be evaluated regarding performance related to metrics associated with the transaction times. Hence, the two transaction types, query and store, can be compared with other related technologies, such as Blockstack [115] and a PostgreSQL database [290] with column-level encryption. Since PRESERVE DNS is a blockchain system that aims to secure the DNS infrastructure, a directly related in-context system is Blockstack. Additionally, a typical query in the literature related to the utilisation of private-permissioned blockchains is if this type of blockchain is more efficient than traditional databases such as PostgreSQL. PostgreSQL offers column-level encryption similar in-context with the Private Data Collection feature of Hyperledger Fabric [66]. DNSTSM [116] is another blockchain system directly related to PRESERVE DNS. However, since their architecture is developed using an older version of Hyperledger Fabric, the crucial Private Data Collection feature is not present; hence only a security comparison is feasible, as presented in Table 3.1.

As illustrated in Table 3.2 and Figure 3.4, the performance metrics to read and write data to the compared systems can be seen. The technical specifications of the system that hosts the blockchain system are as follows: 6th Generation 2.0GHZ dual-core

Table 3.1: Comparison of Methods [66].

Method	*Attack Thwarting*	*User privacy*	*Existing DNS infrastructure*
DecDNS Liu et al. [282]	✓	X	✓
Liang et al. [113]	✓	X	✓
Namecoin Kalodner et al. [114]	✓	✓	X
Blockstack Ali et al. [115]	✓	✓	X
DNSTSM Yu et al. [116]	✓	X	✓
PRESERVE DNS	✓	✓	✓

Intel Core i5 CPU, with 8GB RAM running at 1866 MHz and 256GB PCIe-based flash storage. The amount of data is split into batches of 10, 1000, 10,000, 100,000, and 1,000,000 records to compare PRESERVE DNS using CouchDB, Blockstack using Gaia decentralised storage [266], and a PostgreSQL database with column-level encryption. PostgreSQL is the quicker system to query a smaller number of DNS records with a linear increase to the query times concurrently with the increase of the total stored records. A similarity with DNS records stored in Blockstack's Gaia decentralised storage method is that they are stored off-chain in key-values pairs, similar to the PRESERVE DNS. However, the query times in PRESERVE DNS is quicker than Blockstack and remain the same despite the stored DNS records increase. The number of stored DNS records in production environments usually consists of millions of DNS records; hence, PRESERVE DNS advantages could be further highlighted.

Table 3.2: Read Data/Write Data transaction time in milliseconds (ms) per number of DNS entries [66].

Number of DNS Entries		**10**	**1000**	**10,000**	**100,000**	**1,000,000**
PRESERVE DNS	*Read Data*	180ms	180ms	180ms	180ms	180ms
	Write Data	230ms	230ms	230ms	230ms	230ms
PostgreSQL Database	*Read Data*	2ms	3ms	10ms	44ms	220ms
	Write Data	4ms	5ms	6ms	9ms	11ms
Blockstack Ali et al. [266]	*Read Data*	360ms	360ms	360ms	360ms	360ms
	Write Data	530ms	530ms	530ms	530ms	530ms

To conclude the performance evaluation of PRESERVE DNS, the CPU and Memory usages benchmarked, with the results presented in Figure 3.5 and 3.6. The CPU benchmarks revealed that reading data from the presented system is efficient and consume

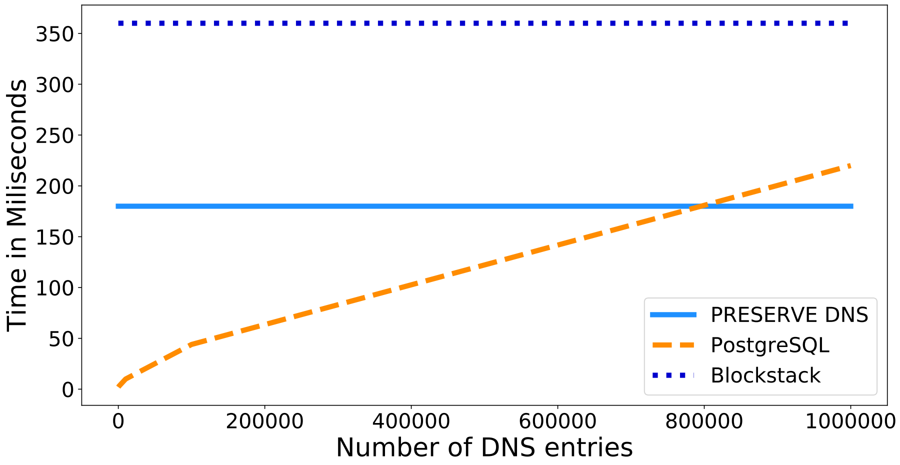
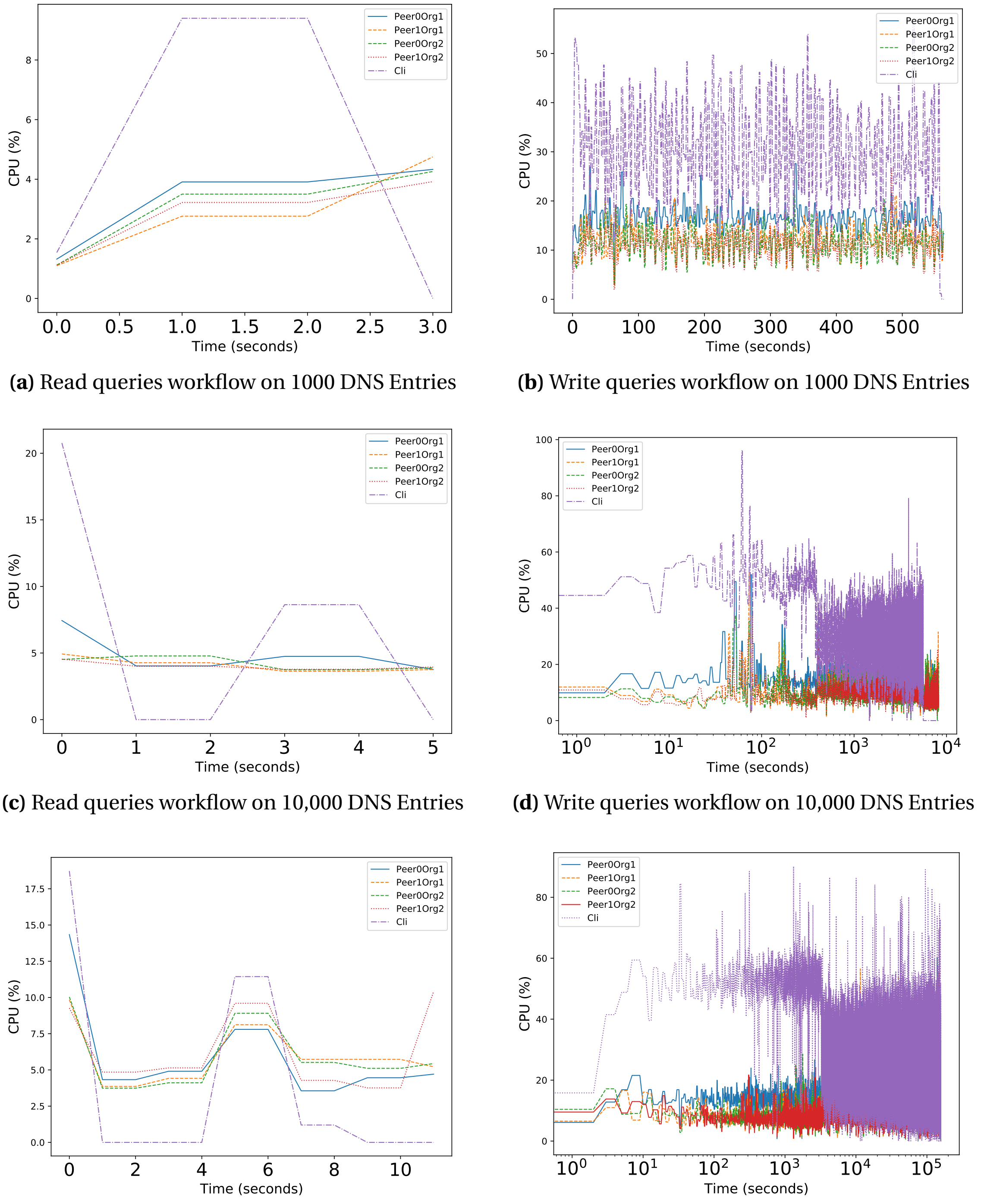


Figure 3.4: Read Data transactions overhead [66].

5%-10% of the CPU in diverging numbers of DNS records, such as 1000, 10,000, 100,000 DNS records. Similarly, writing data to this blockchain system requires less than 20% CPU usage. The CPU benchmarks present each blockchain node's usage, which are identified as Peer0, Peer1 of Organisation 1 and 2, respectively. As seen previously, all the blockchain nodes are in the form of docker containers. The last entity benchmarked, named CLI, is the docker container that is used to utilise the Hyperledger Fabric's command-line interface and transmit the DNS records and blockchain commands. The CPU usage of CLI fluctuates quickly, which is expected since it is the first point of interaction with the rest of the blockchain system. Furthermore, the memory utilisation of the blockchain's nodes for the diverging number of records (1000, 10,000, 100,000 DNS records) is very low, with its minimum and maximum values presented in Figure 3.6.

3.3.3 Summary

DNS is a fundamental infrastructure of the modern internet, and however, a complete transformation of it is not feasible right now. It has developed without security in mind and contains various security gaps that adversaries may exploit. Hence, the systems that aim to secure DNS should ensure there are no conflicts and its procedures normally occur without interruptions. PRESERVE DNS is a blockchain system that can be built on top of the existing DNS infrastructure, adding novel security mechanisms with sufficient performance. As seen in the previous sections, the security and performance



(a) Read queries workflow on 1000 DNS Entries

(b) Write queries workflow on 1000 DNS Entries

(c) Read queries workflow on 10,000 DNS Entries

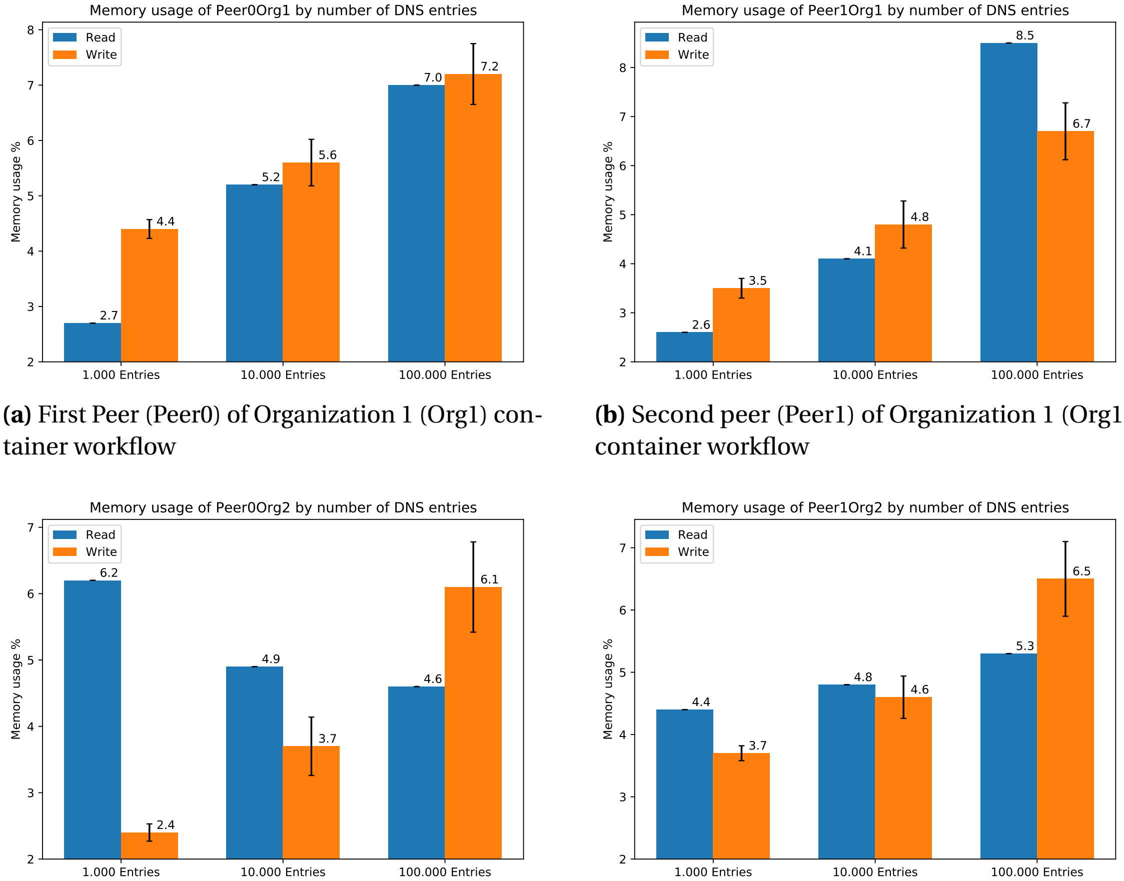
(d) Write queries workflow on 10,000 DNS Entries

(e) Read queries workflow on 100,000 DNS Entries

(f) Write queries workflow on 100,000 DNS Entries

Figure 3.5: CPU Usage (%) of Nodes during workflow [66].

evaluation of PRESERVE DNS revealed that other related in-context systems lack at least one security feature and are often more computationally expensive and gradual than this system. A matter of great importance is that PRESERVE DNS is flexible and can work in combination with other DNS security mechanisms, such as EXPOSURE [91], Notos [90], and Khalil et al. [92], addressing many of their security and privacy flaws



(a) First Peer (Peer0) of Organization 1 (Org1) container workflow

(b) Second peer (Peer1) of Organization 1 (Org1) container workflow

(c) First Peer (Peer0) of Organization 2 (Org2) container workflow

(d) Second Peer (Peer1) of Organization 2 (Org2) container workflow

Figure 3.6: Memory Usage (%) of Nodes during workflow on 1,000, 10,000 and 100,000 DNS Entries [66].

such as the correlation of the internet history and visited websites with the IP addresses of the individuals that performed the DNS queries.

3.4 Case Study: A Privacy-Preserving Electronic Health Records management system

The protection of DNS records is very important, but a demonstration of the previously presented system in an even more critical use case, such as healthcare, illustrates all the benefits of this technology. The practical and adequate protection of EHR is a commonly discussed topic in the literature due to the sensitivity of the data [120]. Nevertheless, as seen previously, a novel technology such as blockchain can ensure the privacy of the data subjects whilst providing adequate performance. This system is PREHEALTH [67], a decentralised EHR management system built using Hyperledger Fabric distributed ledger technology and Idemix, with novel security features to preserve the security of the stored records efficiently.

PREHEALTH's overview can be seen in Figure 3.7 in which different participants such as doctors and hospitals can partially view some of the stored records according to their role (visualised as "Authorized Participants"), whilst the data subjects, the patients, that stored their data in the blockchain system can view all their stored data (visualised as "Users"). Additionally, in this scenario, other governmental bodies may be introduced, such as auditors. The role of the auditors in this scenario could be to monitor information related to a specific healthcare institution (i.e. the number of registered patients, the number of appointments performed in a given time frame, the number of given prescriptions) and that it follows the proper operating procedures and no information about the individual patients or their diagnoses and treatments. Hence, they have the least access privileges. It should be noted that unauthorised participants cannot access any stored information. This scenario is not the only one that can be developed using this infrastructure. Due to its flexibility, more complex designs can be formulated, such as those that particular patients' doctors can only write to the distributed ledger, whereas the patients and other doctors can only read data from it.

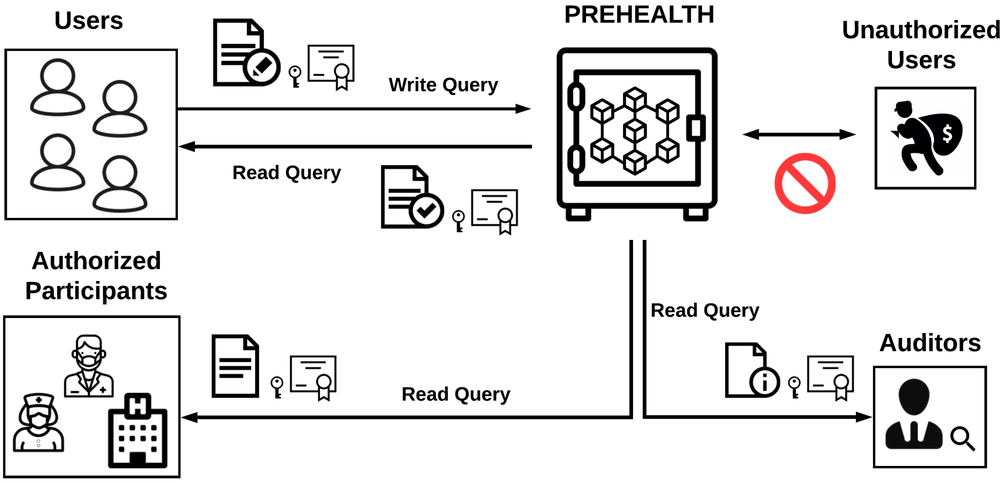


Figure 3.7: PREHEALTH overview [67].

PREHEALTH [67] is an EHR management use case with similar architecture and characteristics that have been initially developed and presented in the work of Papadopoulos et al. [66], such as the private data collection feature.

3.4.1 Architecture and Implementation

PREHEALTH [67] is built using Hyperledger Fabric distributed ledger technology; hence, for the demonstration of the proof-of-concept, the system's participants are in the form of docker containers, developed on top of Debian 9.11 *Stretch* operating system. Following a similar approach to PRESERVE DNS [66], the commands to and from the blockchain system pass through the CLI in order to transmit to the rest of the blockchain nodes, such as peers and orderers. Another similarity with PRESERVE DNS is that the chosen databases among the peers are instances of CouchDB [275, 13] since they offer rich queries to the stored records. An illustration of the PREHEALTH's system overview can be seen in Figure 3.8. However, a fundamental difference with PRESERVE DNS is the creation of identity certificates. In PREHEALTH, Idemix technology has been utilised to create X.509 digital certificates with zero-knowledge proof functionality, such as that no participating entity (e.g. the MSP) cannot reveal sensitive information about the holders of the certificates [67].

The Hyperledger Fabric architecture and the blockchain entities can be seen in Figure 3.9. There are three different participating entities in this scenario, with each one simulating a different healthcare institution. Additionally, each healthcare institution comprises of three blockchain peers and contains its own MSP to handle their identities. Further, there is a cluster of orderers which consists of three identical orderers to handle each blockchain interaction and improve the system's performance and fault-tolerance, illustrated as the Ordering Service. The public blockchain ledgers are available to all the participating organisations, with a variation to Organisation 1, which also holds a private ledger, utilising the Private Data Collection feature. The process that the system follows is that:

- The peer that wants to submit a transaction to the ledger first verifies its identity to its MSP and then sends the transaction to the orderer.

- The orderer checks the transaction's validity, executing it according to the installed chaincode, and if it is verified, creates the new blockchain block.
- The orderer broadcasts the ledger updates to all the participating peers to update their own ledgers.

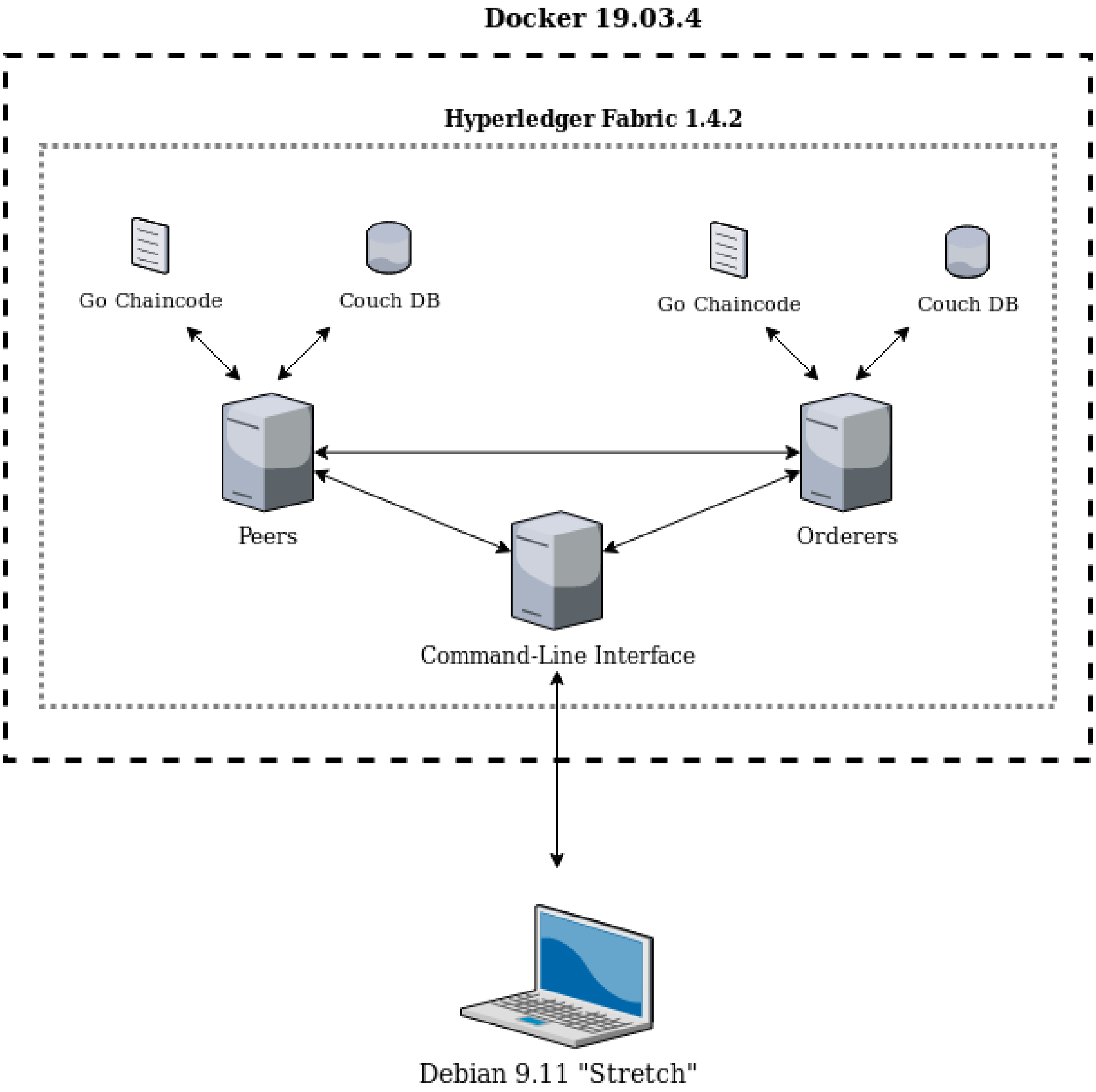


Figure 3.8: PREHEALTH internal technical architecture [67].

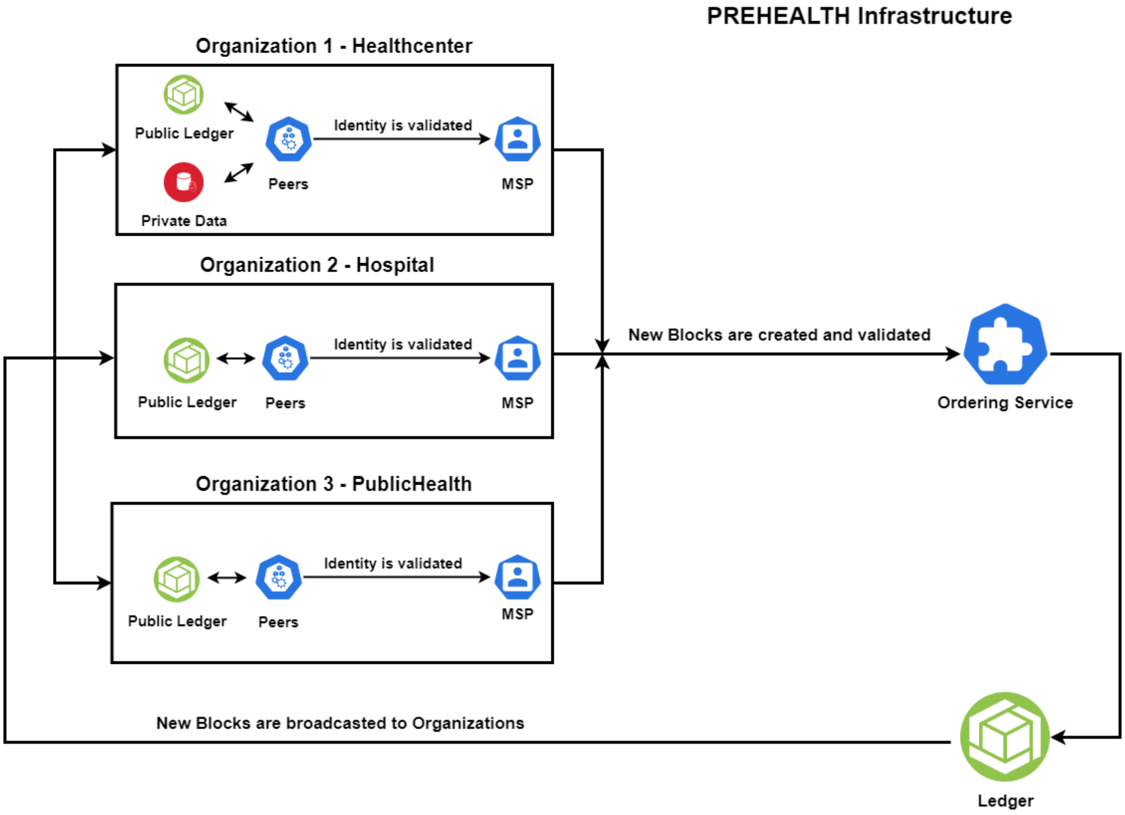


Figure 3.9: PREHEALTH Hyperledger Fabric architecture [67].

3.4.2 Evaluation and Discussion

3.4.2.1 Security Evaluation

As seen in Section 3.3.2.1, since the underlying technology is the same between PRESERVE DNS [66] and PREHEALTH [67], there are similar potential risks to the infrastructure, such as the human factor. In situations the identity certificates are compromised, an unauthorised participant may query data from the blockchain ledger. However, the unauthorised data storage can be potentially prevented, in real-world scenarios, in which the consensus mechanism is carefully set to consider this type of attack, such as a PBFT algorithm [26, 13]. Similarly to PRESERVE DNS, in PREHEALTH, no entity completely controls the blockchain ledgers, not even its administrators.

To evaluate the anonymity of the stored EHR and unlinkability between them, the Client Identity Library (CID) has been used [291]. This library acts as a tool to identify leaked information during blockchain interactions and reveals that the blockchain transactions are transmitted encrypted using Transport Layer Security (TLS), signed by a “cryptographic public key” [13]. Further, as seen in the work of Thakkar et al. [275], the authors present a set of parameters that should be carefully considered in any Hyperledger Fabric deployment to preserve the data subject’s anonymity and unlinkability. This set of parameters is comprised of Static and Dynamic variables and analysed as follows:

Static variables:

- The number of blockchain participating entities, such as the number of organisations, peers and orderers.
- The endorsement policy - the number of participating peers and organisations that need to accept a transaction to be considered valid.

Dynamic Variables:

- The management of the identity certifications - In PREHEALTH the identity certificates are X.509 identity certificates [271], generated through the Idemix technology.
- Users’ registration scheme - As seen previously in Figure 3.8, the Idemix arguments are transmitted through the CLI to manage the users’ registration and their interaction with the blockchain system.

3.4.2.2 Performance Evaluation

To evaluate the performance of PREHEALTH, similar performance metrics were taken, as in PRESERVE DNS. The performance metrics include an EHR management system in a traditional PostgreSQL database with column-level encryption to simulate the Private Data Collection feature, and two other blockchain systems, MedRec and Blockstack, the one directly related to the EHR management, and the other with the potential to develop a decentralised application with similar features. It should be noted that the comparison with MedRec occurred in a proof-of-concept utilising the Proof-of-Authority consensus mechanism, opposed to the commercial version of MedRec that utilises the Proof-of-Work consensus mechanism [135]. The experimentation occurred in varying numbers of EHR (such as 10, 100, 1000, 10,000, 100,000, 1,000,000). Table 3.3 presents the results of the comparison with an illustration following in Figure 3.10. As seen from the table and the figure, PREHEALTH is more efficient than the other blockchain systems and offers quicker transaction times. The data read transaction times using a PostgreSQL are quicker than PREHEALTH for all the tested numbers of EHR (up to 1,000,000 records); however, as seen Figure 3.10 the data read transaction time increases exponentially, with a speculation to surpass PREHEALTH data read transaction times at approximately 1,200,000 EHR. Since the number of stored records consists of millions in a real-world EHR management system, it is appropriate to consider PREHEALTH as a more efficient system in the presented scenario.

Table 3.3: Query time measurements in milliseconds (ms) per number of entries [67].

Number of Records:		**10**	**100**	**1000**	**10,000**	**100,000**	**1,000,000**
PREHEALTH	Read	183 ms	183 ms	183 ms	183 ms	183 ms	183 ms
	Write	58 ms	58 ms	58 ms	58 ms	58 ms	58 ms
PostgresSQL Database	Read	1.73 ms	1.79 ms	2.38 ms	8.76 ms	43.52 ms	136.19 ms
	Write	4.32 ms	4.48 ms	4.47 ms	4.37 ms	4.39 ms	4.45 ms
MedRec [135]	Read	177 ms	186 ms	194 ms	199 ms	205 ms	210 ms
	Write	81.5 ms	86.9 ms	79.6 ms	71.6 ms	63.2 ms	79.6 ms
Blockstack [266]	Read	360 ms	360 ms	360 ms	360 ms	360 ms	360 ms
	Write	530 ms	530 ms	530 ms	530 ms	530 ms	530 ms

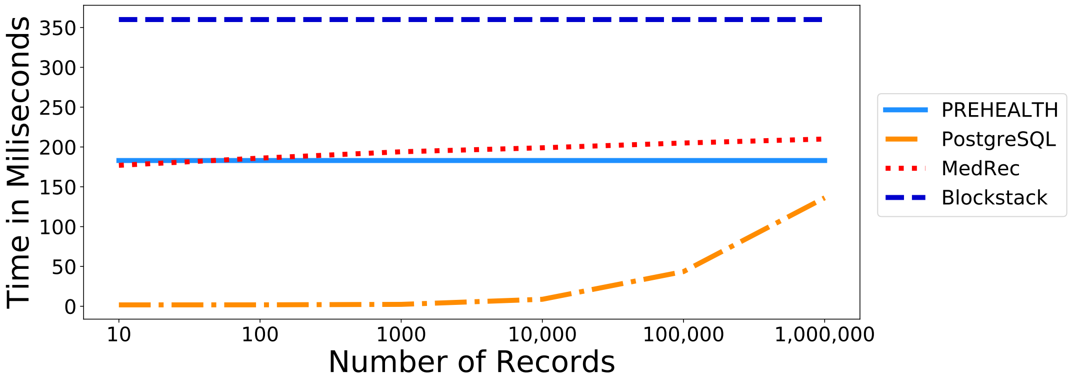
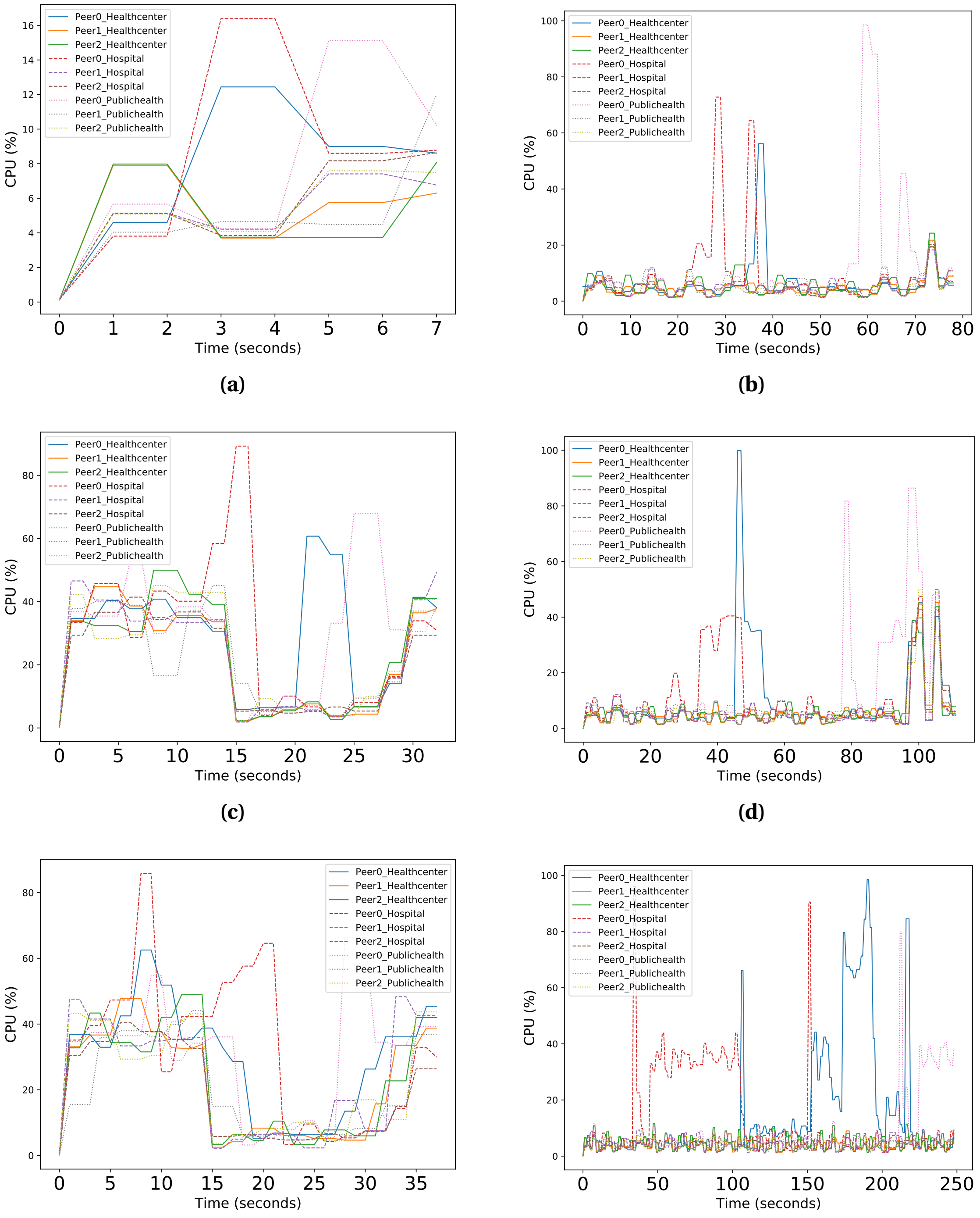


Figure 3.10: Read records transactions overhead [67].

To further evaluate the performance of PREHEALTH, CPU and memory benchmarks have been conducted. The memory benchmarks showed a negligible memory utilisation (less than 2%); hence, they are not illustrated in a plot. The CPU benchmarks during data read and write transactions can be seen in Table 3.4, and illustrated in Figure 3.11. PREHEALTH tested in varying numbers of EHR (such as 1000, 10,000, 100,000 records), monitoring all the three peers of each organisation, namely Peer 0, Peer 1, and Peer 2, of Healthcenter, Hospital, and PublicHealth organisations. As it can be seen from the table and the figure, the CPU utilisation of data read transactions is low (less than 30%), with the data read transactions achieving 15% CPU utilisation. Hardware limitations likely cause the presented CPU utilisation fluctuations; hence, their quick recovery times. The infrastructure's technical specifications were a 2.4GHZ quad-core Intel Core i7 6th Generation CPU, with 8GB RAM and 256GB SSD [67].

Table 3.4: Average CPU (%) performance of each blockchain peer per number of electronic health records.

PREHEALTH Organizations	**PREHEALTH Peers**		**Number of Records**		
			1000	**10,000**	**100,000**
Healthcenter	Peer 0	Read Queries	7.6%	28.7%	29%
		Write Queries	6.7%	10.3%	15.4%
	Peer 1	Read Queries	5.1%	21.8%	21.7%
		Write Queries	4.9%	6.7%	4.2%
	Peer 2	Read Queries	4.9%	23.3%	22.2%
		Write Queries	5.4%	6.4%	4.3%
Hospital	Peer 0	Read Queries	8.3%	29.4%	32.3%
		Write Queries	9.3%	11.2%	13.9%
	Peer 1	Read Queries	5.1%	22.7%	23.2%
		Write Queries	5.4%	6.4%	4.3%
	Peer 2	Read Queries	5.4%	20.7%	18.7%
		Write Queries	4.9%	6.6%	4.2%
PublicHealth	Peer 0	Read Queries	7.6%	30.5%	30.3%
		Write Queries	11.4%	12.8%	8.2%
	Peer 1	Read Queries	4.8%	22%	20.1%
		Write Queries	5.3%	6.8%	4%
	Peer 2	Read Queries	5.1%	23.4%	22%
		Write Queries	4.7%	6.6%	4%



(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 3.11: (**a**) Read queries workflow on 1000 Records. (**b**) Write queries workflow on 1000 Records. (**c**) Read queries workflow on 10,000 Records. (**d**) Write queries workflow on 10,000 Records. (**e**) Read queries workflow on 100,000 Records. (**f**) Write queries workflow on 100,000 Records. CPU Usage (%) of blockchain peers during workflow [67].

3.4.3 Summary

The challenge of the EHR management and storage is critical due to the data sensitivity and further intensified due to privacy regulations, such as the GDPR [4]. Systems that aim to secure these functions and solve these challenges should build with security principles in mind aiming to protect the stored data from multiple perspectives. Blockchain technologies are often considered inappropriate to store highly sensitive data; however, specific distributed ledger technologies can store this data efficiently protected and ensuring its privacy by strong cryptographic mechanisms. A system with all these advantages is PREHEALTH, providing all the necessary security and privacy guarantees, as well as approaches for its further scalability, auditability, and immutability to a real-world infrastructure. As presented in the previous sections, PREHEALTH is also performance efficient, providing suitable CPU and memory computational overheads.

3.5 Conclusion

This chapter focused on the assurance of stored records in critical infrastructures through blockchain [66, 67]. A system was developed and presented, namely, PRESERVE DNS [66], that can efficiently protect the stored data, presenting novel data storage and query functions, such as the Private Data Collection, built on top of a distributed ledger technology. This chapter presents extensive security and performance evaluations of this system, comparing it to other related works in the literature and concluding with valuable outcomes. This system aims to combat common issues related to the privacy leakage of the data subjects since their IP addresses are considered personal data, protecting against potential domain records misuse and DNS profiling. It should be noted that the securely stored passive DNS records in PRESERVE DNS [66] can be further analysed and monitored automatically through ML techniques to identify potential phishing URLs quickly and efficiently (Appendix C) [43]. To further demonstrate the system's flexibility, adaptability, and security, a healthcare case study was presented that is focused on the management of highly-sensitive EHR, namely PREHEALTH [67]. As previously, extensive security and performance evaluations were presented in comparison with other related works. A matter of great importance is that since the distributed ledger technology's fundamental is its private-permissioned nature, identity credentials play a crucial role in the infrastructure. Hence, the novel Idemix technology utilised in PREHEALTH provides the necessary zero-knowledge proofs to enhance the system's privacy further.

Potential future avenues for the presented infrastructures include incorporating other self-sovereign identity systems that offer certificates revocation and digital attributes authentication such as Decentralised Identifiers (DIDs) [292] and Verifiable Credentials (VCs) [293] utilising the Hyperledger Aries framework [294]; with a few suggestions following in the next chapter. Utilising these technologies, certificate management could occur through end-to-end encrypted systems, enhancing the total security and privacy of the infrastructure.

Fourth Chapter

Digital Identities and Privacy-Preserving Machine Learning

4.1 Introduction

This chapter addresses the **Objective III** by developing an architecture built using SSI technology combined with other privacy-preserving technologies, such as a blockchain identity ledger and privacy-preserving machine learning. This concept indicates that the system's participants have sole control of their identities and can regulate which of their data the other participants are allowed to access. Each participant utilises DIDs [292] and VCs [293] similarly to a Public-Private Key Infrastructure [295, 296].

The emerging concept of SSI is being developed by various open-source or commercial standards, with a common aspect, all of them are citizen-centric and aim to protect the citizens' privacy. Utilising SSI technologies, the digital identities are controlled solely by their owners, the citizens, without requiring trusted intermediaries. Additionally, due to the data sensitivity in machine learning analysis, developed techniques aim to address many of its privacy issues. These privacy-preserving machine learning approaches aim to protect citizens from pervasive analysis of their sensitive data. Hence, combining these technologies would transfer the control back to its data subjects, the citizens, to control which data they want to share, with whom, and for how long, in a privacy-preserving manner.

The SSI technologies were firstly developed to accommodate functionalities, such as the presentation of credentials and proofs, as well as basic messaging between participants. The establishment of trust among the parties occurs through the presentation of credentials and proofs that mutually trusted authorities have issued. The party that holds the legitimate credential or proof is using a form of a digital wallet to store it, and the party that verifies the credential or proof inspects an identities database, often a distributed ledger, to confirm its validity. Due to its complex nature, the technological advancements focused on developing perplexing identities scenarios mimicking real-world problems. Such advancements focus on the credentials' issuing, holding, and verification in real-world environments. However, using these SSI technologies on top of other data analysis techniques such as ML remained obscure by the research community due to their own introduced complexities and challenges. A matter of great importance is also that combining these technologies should be as pleasant as possible for the users without irritating the users with additional gratuitous intricacies.

4.2 Self-Sovereign Identity Foundations

4.2.1 Decentralised Identifiers

DIDs confirmed as an accepted digital identifier to establish trust in distributed systems during a working group [292] of the World Wide Web Consortium (W3C). Only the owner of each DID manages it and can allow another participating entity to utilise it to verify its authenticity. Hence, participants can use the DIDs novel characteristics to log in to a system without trusting a third-party company but directly verifying the DID owner. Since DIDs need to be verified by other participants, it is common to be stored in distributed ledgers and blockchain identity management schemes [297]. The methods to verify a DID vary according to the utilised underlying technology [298]. An overview of DID documents, as follows:

- ID field — the ID that resolves the specific DID document;

- Public key;
- Authentication protocols; and
- Service endpoints

In Listing 2.1, the resolution of *did:example:123456789abcdefghi* DID document using the DID method *example* and the identifier *123456789abcdefghi* can be seen [68].

Listing 4.1: .]An example DID document [68].

```
{
  "@context": "https://example.org/example-method/v1",
  "id": "did:example:123456789abcdefghi",
  "publicKey": [{
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "RsaVerificationKey2018",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
  }],
  "authentication": [
    "did:example:123456789abcdefghi#keys-1",
 ],
  "service": [{
    "id": "did:example:123456789abcdefghi#agent",
    "type": "AgentService",
    "serviceEndpoint": "https://agent.example.com/8377464"
  }]
}
```

Utilising DID specifications, the interoperability is ensured between various resolution schemes, agnostic of the used underlying storage infrastructure. Additionally, another method for participants to interact utilising DIDs is via Peer DIDs [299], in which each one of them sustains their own list of DID documents, and no storage scheme is required for their resolution.

4.2.1.1 Decentralised Identifiers Communication Protocol

Another Hyperledger's open-source project that utilises DIDs is Hyperledger Aries [294]. Hyperledger Aries exchanges DIDs through the end-to-end encrypted DID Communication (DIDComm) protocol that is similar to the work of Chaum [300]. Decentralised Identity Foundation [301] is developing the DIDComm protocol. DIDComm is an end-to-end encrypted, asynchronous protocol that can interpret some information of the DID document, such as the recipient's endpoint address and the public key, to verify the participants' integrity and authenticity, and exchange private and secure messages.

In the scenario of two participants who want to communicate, such as Alice and Bob, they need to sign their messages with their keys, so the other participant can utilise DIDs to verify their identity. For example, Alice encrypts and signs with her private key a message that is intended for Bob. Then before Bob decrypts and reads the message, he needs to check the message's integrity by verifying Alice's public key using a public record, such as a blockchain identity management scheme [297]. Bob can also check that Alice holds a credential provided by a legitimate authority. In that case, Bob adds Alice adds to his list of verified-trusted contacts. On the other end, when Bob sends a message to Alice, she needs to perform the same authentication and verification check, and if Bob holds a credential issued by a legitimate authority, she can add him to her list of trusted contacts. Each participant's DID document encapsulates all the required information to fulfil a DIDComm interaction, such as the mentioned example. So, after this mutual authentication and verification, the two participants can utilise the established DIDComm channel to further securely exchange data and messages [68], as it can be seen in Algorithm 1. The various encryption techniques used by the DIDComm protocol and guarantee its privacy and security include elliptic curve [302], Rivest–Shamir–Adleman (RSA) [303] and ElGamal [304] techniques.

4.2.2 Verifiable Credentials

Another set of tools the participating parties can use to verify the other party's identity are VC. These participating entities consist of the *Issuers* entity that issues a VC to a

Algorithm 1 DID Communication Between Alice and Bob [68].

```
Alice holds her private key $sk_a$ and Bob's DID Document that includes his endpoint address ($endpoint_{bob}$) and his public key ($pk_b$).
Bob holds his private key ($sk_b$), and Alice's DID Document that includes her public key ($pk_a$).
Alice encrypts a plaintext message ($m$) using Bob's public key ($pk_b$) and creates an encrypted message ($e_b$).
Alice signs the encrypted message ($e_b$) using her private key ($sk_a$) and creates a signature ($\sigma$).
Alice sends ($e_b$,$\sigma$) to $endpoint_{bob}$.
Bob's endpoint ($endpoint_{bob}$) receives the message from Alice.
Bob verifies $\sigma$ using Alice's public key $pk_a$.
if Verify($\sigma, e_b, pk_a$) = 1 then
    Bob decrypts $e_b$ using his private key ($sk_b$).
    Bob reads the plaintext message ($m$) sent by Alice.
end if
```

Holder, and the *Verifiers*, that verify the authenticity of the *Holder*'s VC [293], as it can be seen in Figure 4.1. As in DIDs, it is common to utilise a blockchain identity management scheme to store the VC securely [297].

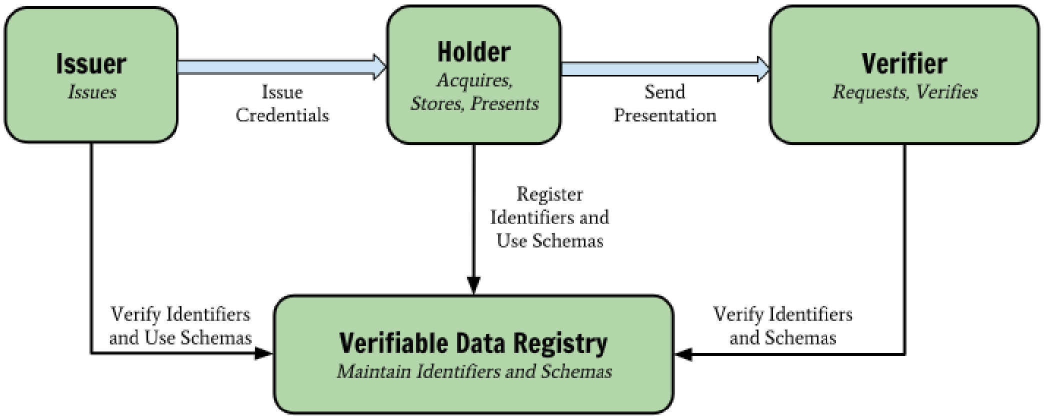

Issuer
Issues
Issue Credentials
Holder
Acquires, Stores, Presents
Send Presentation
Verifier
Requests, Verifies
Register Identifiers and Use Schemas
Verify Identifiers and Use Schemas
Verifiable Data Registry
Maintain Identifiers and Schemas
Verify Identifiers and Schemas

Figure 4.1: Verifiable Credential Roles [68, 293].

In order for the *Issuer* to issue a new VC, they need to generate a signature from their private key that resolves the public key of their DID document. The supported generated signature scheme can be one of the Linked Data signatures [305], JSON Web signatures [306], or finally, Camenisch-Lysyanskaya (CL) signatures [307, 308].

The Hyperledger Aries, in order to enhance the system's security without authentication similarly to the work of Chaum [309], utilises CL signatures by creating a blinded link secret that is bonded to their participating entities without the VC *Issuers* have knowledge of the secret values.

The *Verifiers* entity accept the received VC after they confirm the following:

1. The *Issuer*'s ID can be resolved to a DID document stored on a public ledger, such as a blockchain identity management scheme [297]. Additionally, the *Verifiers* need to ensure that the DID document contains a public key that they can use to sign the VC for integrity purposes.

2. The VC *Holder* can create a Zero-Knowledge Proof (ZKP) to demonstrate and prove the blinded linked secret.

3. The *Issuer* has the permission to issue this kind of VC. Since the *Verifiers* accept only *Holders* that poses a VC from a legitimate *Issuer*, they eliminate the possibility of fraud. Hence, it is possible to create legal documents that describe the system's operating capabilities [310].

4. The legitimate *Issuer* has not revoked the presented VC. It is possible for a *Holder* to poses a VC that is no longer valid. This check can be done by checking that a revocation registry, such as a cryptographic accumulator [311], for the specific VC has not been filed on the public ledger.

5. Last but not least, the *Verifier* needs to verify that the attributes included in the VC meet the infrastructure's authorisation criteria. A VC can be valid only for a certain period and then lose its validity.

All the communication in the above example occurs via the encrypted DIDComm protocol. The importance of the VC is significant since the participating entities actions can be automated and implemented in large-scale infrastructures since the *Verifiers* can resolve and verify the DID document themselves, without the need to contact the VC's *Issuer* directly.

4.3 Private and Trusted Federated Machine Learning

Traditional machine learning faces challenges due to adversarial behaviour. Luckily, there is a privacy-preserving machine learning field that is focused on combating many of these issues. More specifically, FL is one of the most prominent technologies of this field, which distributes the training of ML algorithms to the participants. However, the distribution of the ML computation introduces new challenges and risks. FL itself does not guarantee protection against adversarial attacks that aim to manipulate the ML algorithms, such as data and model poisoning, or adversarial intermediary attacks, such as Man-In-The-Middle (MITM) attacks. Nevertheless, if FL is combined with other privacy-preserving SSI techniques, the participants can establish trust and enhance the system's privacy.

In the following sections, a privacy-preserving architecture is presented, namely Trusted Federated Learning (TFL) [68], that merges SSI and FL to create a unified, trusted ecosystem, within a healthcare scenario, with authentication and verification checks required to train ML algorithms distributed. Additionally, all the authentications, verifications, and data transmission occur within end-to-end encrypted communication channels to ensure the protection of this workflow. TFL is evaluated in terms of security guarantees, ML accuracy and computational performance, and these metrics are presented in the following sections.

4.3.1 Architecture and Implementation

This subsection presents the technical architecture, implementation and all the technical details followed to create the proof-of-concept. The information presented include the trust establishment, the utilised end-to-end encrypted communication protocol, and the FL process.

A healthcare ecosystem was developed that simulates three NHS hospitals and a researcher aiming to train a ML model using hospital-held private data. The aspect of trust is critical within such highly-sensitive environments. Within this ecosystem, the hospitals and the researcher need to perform mutual authentication and verifications tests to ensure that the other party holds a legitimate credential issued by a regulatory authority or the NHS Trust. An overview of the ecosystem can be seen in Figure 4.2.

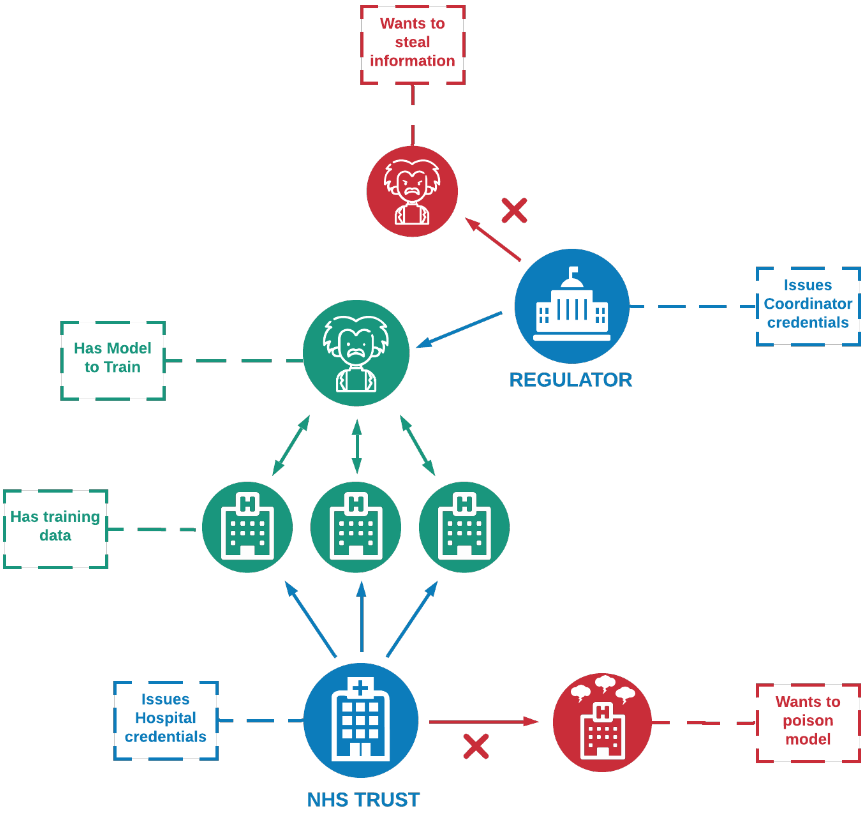

Wants to steal information
Has Model to Train
Issues Coordinator credentials
REGULATOR
Has training data
Issues Hospital credentials
NHS TRUST
Wants to poison model

Figure 4.2: Healthcare trust model overview [68].

The scenario starts when one of the hospitals establishes a connection with the NHS Trust, which issues a VC to the corresponding hospital. Similarly, this process occurs again for all the participating hospitals. For the issuing process, the NHS Trust utilises a public identities blockchain ledger such as the British Columbia VON's development ledger [312], to store the public DIDs. From the researcher's perspective, instead of the NHS Trust, the connection establishment is with a governmental regulatory authority that issues a researcher VC and writes the public DID to the same public identities blockchain ledger. Furthermore, the researcher is able to connect to each hospital to perform the mutual authentication and verification process. During this process, both parties need to present their identity proofs that the other party is checking against the DIDs publicly stored in the identities blockchain ledger. Unauthorised participants cannot infiltrate this ecosystem since their fake VC would not be resolved successfully to the legitimate DIDs. Additionally, all the connections occur within end-to-end encrypted DIDComm channels; hence, malicious parties cannot interfere in-between the connections and perform attacks such as Man-In-The-Middle. For the development of the technical testbed, a step-by-step approach is followed, as seen in Algorithm 2.

Algorithm 2 Establishing Trusted Connections [68].

```
Researcher agent exchanges DIDs with the Regulator agent to establish a DIDComm
channel.
Regulator offers an Audited Researcher-Coordinator credential over this channel.
Researcher accepts and stores the credential in their wallet.
for each Hospital agent do
    Initiate DID Exchange with NHS Trust agent to establish DIDComm channel.
    NHS Trust offers Verified Hospital credentials over DIDComm.
    Hospital accepts and stores the credential.
end for
for each Hospital agent do
    Hospital initiates DID Exchange with Researcher to establish DIDComm channel.

    Researcher requests proof of Verified Hospital credential issued and signed by the
    NHS Trust.
    Hospitals generate a valid proof from their Verified Hospital credential and re-
    spond to the Researcher.
    Researcher verifies the proof by first checking the DID against the known DID
    they have stored for the NHS Trust, then resolve the DID to locate the keys and
    verify the signature.
    if Hospitals can prove they have a valid Verified Hospital credential then
        Researcher adds the connection identifier to their list of Trusted Connections.
    end if
    Hospital requests proof of Audited Researcher credential from the Researcher.
    Researcher uses Audited Researcher credential to generate a valid proof and re-
    sponds.
    Hospital verifies the proof, by checking the signature and DID of the Issuer.
    if Researcher produces a valid proof of Audited Researcher then
        Hospital saves connection identifier as a trusted connection.
    end if
end for
```

Upon successful authentication and verification of all the parties, the researcher can initiate a FL training to send the ML model through the DIDComm channel, encrypted to the recipient hospitals. The hospitals train this model using their private data and send the trained model back to the researcher. The researcher forwards the trained model to the next hospital, which performs the same process until all the hospitals train it. All the participating hospitals have trained the final trained model that the researcher possesses without revealing any sensitive raw information directly to the researcher. The private data derived from a mental healthcare dataset [313] split into four partitions, three training datasets, one for each hospital and a validation dataset possessed by the researcher for testing purposes, in order to evaluate the ML model and measure its accuracy.

All the participants within this ecosystem are configured as Docker containers that mimic computational devices within their own environments. The DIDComm channels add a layer of encryption on top of the transport protocol; in this case, the transport protocol is the Hypertext Transfer Protocol (HTTP) at defined public ports that each participant's agent exposes to the network. An overview of the technical details and ports used can be seen in Figure 4.3 and Table 4.1.

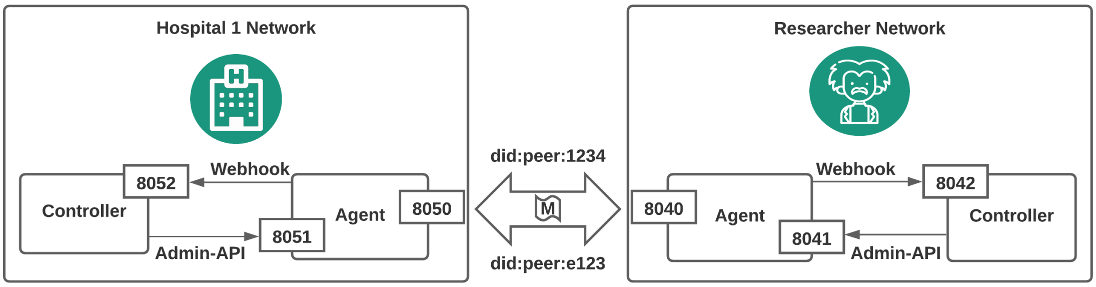


Figure 4.3: Networking communication architecture [68].

Table 4.1: Participating entities communication details [68].

Name	HTTP Port	Admin-API Port	Webhook Port
Hospital 1	8050	8051	8052
Hospital 2	8060	8061	8062
Hospital 3	8070	8071	8072
Researcher	8040	8041	8042
NHS Trust	8020	8021	8022
Regulator	8030	8031	8032

As seen previously, the researcher sends the FL model sequentially to each participating hospital [314, 315, 316, 71]. The step-by-step FL workflow can be seen in Algorithm 3. The combination of SSI and FL is able to combat some common ML threats that derive from the lack of trust between the participants and from the remote training of the ML model. Hence, the hyperparameter tuning of the ML parameters in order to improve the classification accuracy is left out of this experiment's scope. This is a basic FL process that could be improved in real-world architectures by using a more secure variation of FL that introduces a secure aggregator that averages each ML model updates using the Federated Averaging algorithm [71].

Algorithm 3 Federated Learning workflow [68].

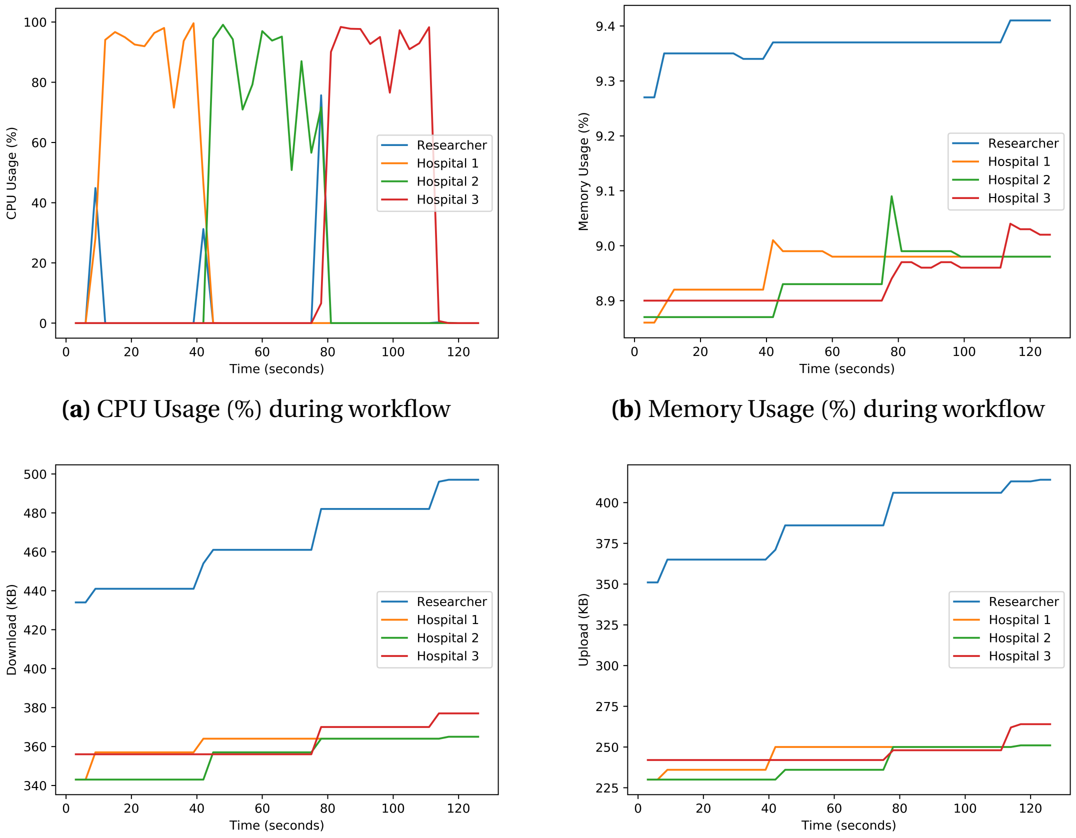
```
Researcher has validation data and a ML model, Hospitals have training data.
while Hospitals have not trained their training data do
    Researcher benchmarks the model's performance against validation data and
    sends the model to the next Hospital.
    Hospital trains the model with their data and then sends the resulting model back
    to the Researcher.
end while
Researcher benchmarks the final model against validation data.
```

4.3.2 Evaluation

A set of performance metrics and security tests have been performed to verify that the implementation works properly. The results of these tests can be seen in the following subsections.

4.3.2.1 Performance Evaluation

To evaluate the performance of the presented infrastructure, a set of benchmark metrics tests were conducted to monitor the CPU, RAM, and network usage. As it can be seen in Figure 4.4a), the CPU usage of the researcher's agent is being raised in every sequence it sends the ML model to the corresponding hospital, followed by a rise in the CPU usage of that hospital that trains it using its private data. Similarly, as seen in Figures 4.4b–d)the monitored RAM and network usages adhere to the previous results as expected, with a small difference that the researcher agent is validating the ML model after every training batch; hence, its RAM and network usages gradually increase. Furthermore, a comparison followed to benchmark the agents' performance when the ML model is being transmitted through the DIDComm channel and without through standard ML training. The results of these tests can be seen in Figures 4.5a) and b). The RAM and network usages follow a similar pattern as the Figures 4.4b–d, hence they have not been visualised.



(a) CPU Usage (%) during workflow

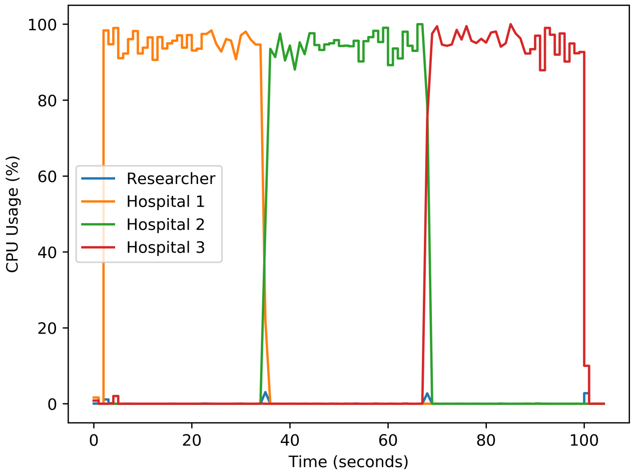
(b) Memory Usage (%) during workflow

(c) Network Input (kB) during workflow

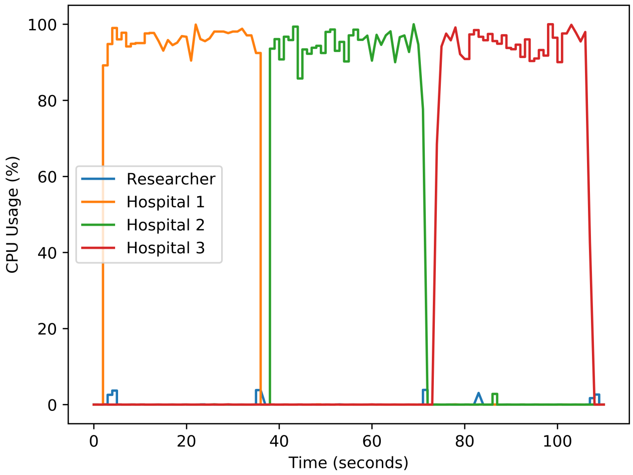
(d) Network Output (kB) during workflow

Figure 4.4: CPU, Memory usage and Network use of Docker container agents during workflow using the original federated learning architecture of [47] [68].

Further tests have been conducted to monitor the performance of the ML model, including its classification accuracy. The experiments were conducted for 10 epochs with a learning rate of 0.01. The dataset is split into four parts (one dataset for each hospital used for training - and one dataset for the researcher used for testing), and its data are being used in batches of 8. Additionally, as mentioned previously, the researcher's agent validates the ML model after each training batch using a testing dataset and produces the confusion matrices shown in Tables 4.2 and 4.3 to confirm that the ML model is being trained successfully. The classification accuracy of the ML model is being calculated by dividing all the outcomes from the correct predictions (the addition of TP and TN) (Appendix B), and the results presented in Tables 4.2 and 4.3,



(a) CPU Usage (%) during workflow and transmission of the model through the DIDComm protocol



(b) CPU (%) during workflow without the use of the DIDComm protocol

Figure 4.5: CPU Usage comparison of Docker containers during workflow using the novel federated learning libraries of 68.

to display the classification accuracy using the Sigmoid linear activation function and the Rectified Linear activation function (ReLu) accordingly [317, 318]. The comparison of the FL classification accuracy through the DIDComm protocol and without it can be seen in Tables 4.4 and 4.5. Finally, as mentioned in the previous section, these experiments aim solely to demonstrate that FL processes are possible through the presented trusted ecosystem. The aspect of the ML training itself is out of the scope of this work. However, after careful hyperparameter tuning, these results could potentially be further improved.

Table 4.2: Classifier's accuracy on the testing dataset without hyperparameters' optimisation over federated learning rounds, using Sigmoid activation function on the original federated learning architecture of [47] [68].

Batch	**0**	**1**	**2**	**3**
True Positives	0	109	120	134
False Positives	0	30	37	41
True Negatives	114	84	77	73
False Negatives	144	35	24	10
Accuracy	44.1%	74.8%	76.3%	80.2%

Table 4.3: Classifier's accuracy on the testing dataset without hyperparameters' optimisation over federated learning rounds, using ReLu activation function [68].

Batch	**0**	**1**	**2**	**3**
True Positives	144	121	121	108
False Positives	0	23	23	36
True Negatives	114	34	33	39
False Negatives	0	80	81	75
Accuracy	100%	60%	59,6%	57%

Table 4.4: Classifier's accuracy on the testing dataset without hyperparameters' optimisation over federated learning rounds through the DIDComm protocol [68].

Batch	**0**	**1**	**2**	**3**
True Positives	0	115	120	135
False Positives	0	29	24	9
True Negatives	113	30	39	44
False Negatives	145	84	75	70
Accuracy	43.7%	56.2%	61.6%	69.3%

4.3.2.2 *Security Evaluation*

Many privacy issues faced in the traditional centralised ML can be mitigated by FL since the ML model is being sent to each data holder [319, 320] instead of data transmitting to a centralised location. However, ML models, even if they are trained and distributed using FL, still suffer from other privacy attacks such as membership inference [183] and model inversion attacks [189, 188, 321]. These attacks elaborate on the scenario of a

Table 4.5: Classifier's accuracy on the testing dataset without hyperparameters' optimisation over federated learning rounds without the DIDComm protocol [68].

Batch	**0**	**1**	**2**	**3**
True Positives	0	113	120	133
False Positives	0	31	24	11
True Negatives	116	35	43	45
False Negatives	142	79	71	69
Accuracy	44.9%	57.3%	63.1%	69%

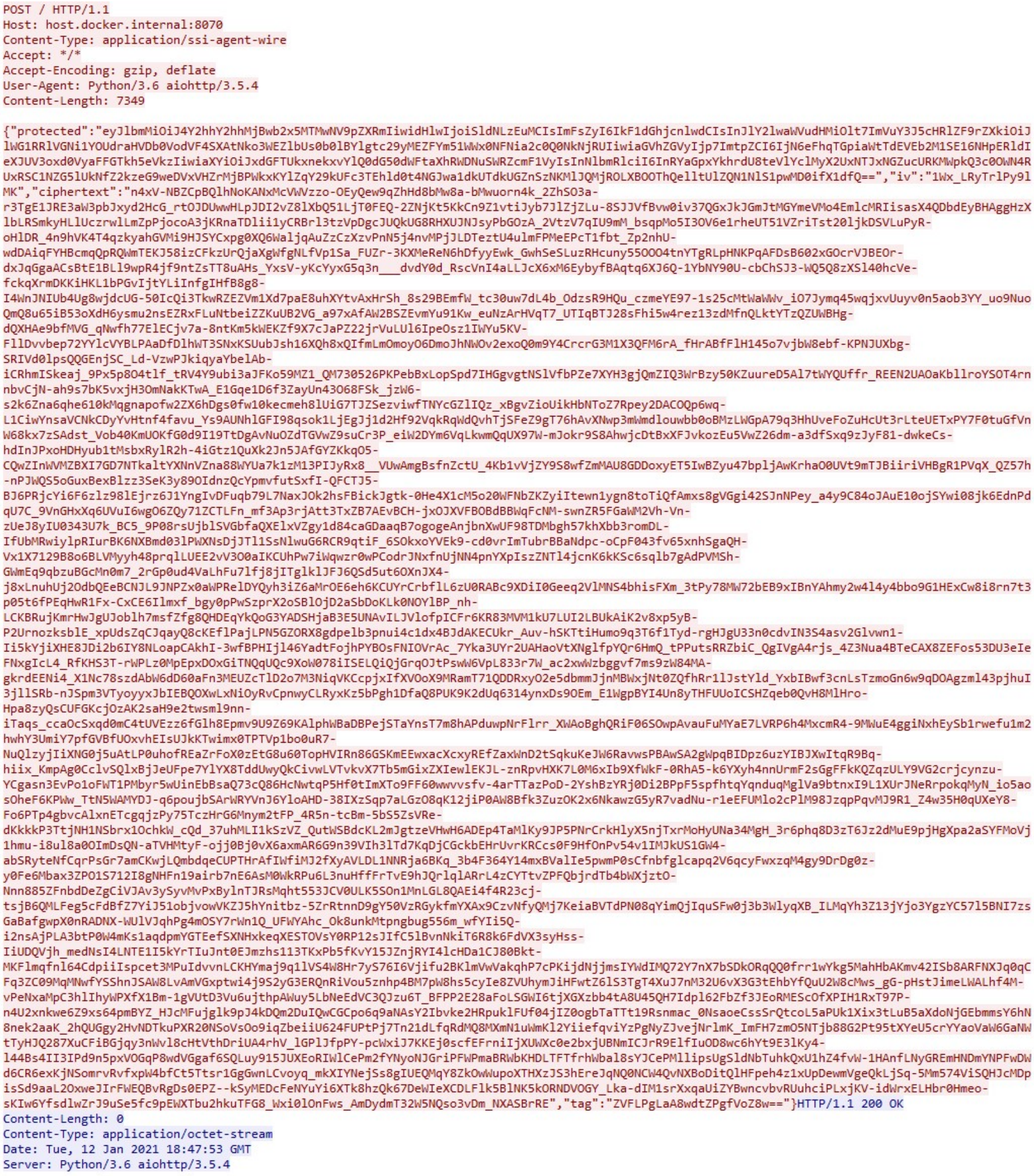
malicious Researcher that could identify if certain data were part of the training, and in the latter attack scenario in which the malicious Researcher could reconstruct a part of the training values [68, 47].

Additionally, since the ML model is being sent to the data holders, the system is exposed to model stealing and model poisoning attacks. In model stealing attacks, a malicious participant stores a copy of the original ML model to avoid paying fees for using it, or to create ML models with similar decision boundaries to profit from them by selling them to other third parties. In model poisoning attacks, a malicious participant inserts backdoors-trojans [322, 323, 213] to the ML model before returning it to the Researcher. It is troublesome for the Researcher to identify if the trained ML model is poisoned since it behaves maliciously only on the backdoor trigger activations. Oppositely to data poisoning attacks [324, 208, 207, 206, 205], in which the ML model's accuracy may drop [204], model poisoning attacks remain hidden.

A number of promising defensive mechanisms against these attacks were presented previously in Chapter 2; however, there are still open problems in FL that are yet to find an effective solution [46].

In the ecosystem presented previously, by using VC, it is able to establish domain-specific trust between the participants. Only mutually authenticated and verified participants can participate in the ML training scenario. Other unauthorised parties are immediately rejected and cannot interfere or interact with the other participants. To evaluate the infrastructure's security, fake Hyperledger Aries agents were created to attempt to participate in the ML training or receive an authorised VC illegally from the NHS Trust or the governmental regulatory authority. The results from these tests showed that any self-signed VC or unauthorised connection requests are immediately rejected, and no participant can interact with the ML training if they do not hold a legitimate VC from an authorised authority [68, 47].

For the security evaluation of the end-to-end encrypted DIDComm messaging protocol, the Wireshark and Tcpdump network packet sniffers were utilised to capture and monitor any ingress and egress network traffic [325]. Since the participants in the ecosystem are in the form of docker containers, the Docker virtual network card was required to be monitored in the host computational device [326]. These results revealed that all the communication between the participants, the proof requests and presentations, the names of the participants and their DIDs are all encrypted. Additionally, each step of the FL training and the ML model transmission are encrypted, and the results can be seen in Figure 4.6. Finally, the Government Communications Headquarters (GCHQ) CyberChef [327] utilised in order to attempt to decode the encrypted content; however, all the attempts were unsuccessful and indicated that the DIDComm protocol is protected, at least, against typical malicious infiltrators. This is very encouraging due to the fact that often the data used for ML training could be sensitive.



```
POST / HTTP/1.1
Host: host.docker.internal:8070
Content-Type: application/ssi-agent-wire
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Python/3.6 aiohttp/3.5.4
Content-Length: 7349

{"protected":"eyJlbmMiOiJ4Y2hhY2hhMjBwb2x5MTMwNV9pZXRmIiwidHlwIjoiSldNLzEuMCIsImFsZyI6IkF1dGhjcnlwdCIsInJlY2lwaWVudHMiOlt7ImVuY3J5cHRlZF9rZXkiOiJ
lWG1RRlVGNi1YOUdraHVDb0VodVF4SXAtNko3WEZlbUs0b0lBYlgtc29yMEZFYm51WWx0NFNia2c0Q0NkNjRUIiwiaGVhZGVyIjp7ImtpZCI6IjN6eFhqTGpiaWtTdEVEb2M1SE16NHpERldI
eXJUV3oxd0VyaFFGTkh5eVkzIiwiaXYiOiJxdGFTUkxnekxvYlQ0dG50dWFtaXhRWDNuSWRZcmF1VyIsInNlbmRlciI6InRYaGpxYkhrdU8teVlYclMyX2UxNTJxNGZucURKMWpkQ3c0OWN4R
UxRSC1NZG5lUkNfZ2kzeG9weDVxVHZrMjBPWkxKYlZqY29kUFc3TEhld0t4NGJwa1dkUTdkUGZnSzNKMlJQMjROLXBOOThQelltUlZQN1NlS1pwMD0ifX1dfQ==","iv":"1Wx_LRyTrlPy9l
MK","ciphertext":"n4xV-NBZCpBQlhNoKANxMcVWVzzo-OEyQew9qZhHd8bMw8a-bMwuorn4k_2ZhSO3a-
r3TgE1JRE3aW3pbJxyd2HcG_rtOJDUwwHLpJDI2vZ8lXbQ51LjT0FEQ-2ZNjKt5KkCn9Z1vtiJyb7JlZjZLu-8SJJVfBvw0iv37QGxJkJGmJtMGYmeVMo4EmlcMRIisasX4QDbdEyBHAggHzX
lbLRSmkyHLlUczrwlLmZpPjocoA3jKRnaTDlii1yCRBrl3tzVpDgcJUQkUG8RHXUJNJsyPbGOzA_2VtzV7qIU9mM_bsqpMo5I3OV6e1rheUT51VZriTst20ljkDSVLuPyR-
oHlDR_4n9hVK4T4qzkyahGVMi9HJSYCxpg0XQ6WaljqAuZzCzXzvPnN5j4nvMPjJLDTeztU4ulmFPMeEPcT1fbt_Zp2nhU-
wdDAiqFYHBcmqQpRQWmTEKJ58izCFkzUrQjaXgWfgNLfVp1Sa_FUZr-3KXMeReN6hDfyyEwk_GwhSeSLuzRHcuny55OO04tnYTgRLpHNKPqAFDsB602xGOcrVJBEOr-
dxJqGgaACsBtE1BLl9wpR4jf9ntZsTT8uAHs_YxsV-yKcYyxG5q3n___dvdY0d_RscVnI4aLLJcX6xM6EybyfBAqtq6XJ6Q-1YbNY90U-cbChSJ3-WQ5Q8zXSl40hcVe-
fckqXrmDKKiHKL1bPGvIjtYLiInfgIHfB8g8-
I4WnJNIUb4Ug8wjdcUG-50IcQi3TkwRZEZVm1Xd7paE8uhXYtvAxHrSh_8s29BEmfW_tc30uw7dL4b_OdzsR9HQu_czmeYE97-1s25cMtWaWWv_iO7Jymq45wqjxvUuyv0n5aob3YY_uo9Nuo
QmQ8u65iB53oXdH6ysmu2nsEZRxFLuNtbeiZZKuUB2VG_a97xAfAW2BSZEvmYu91Kw_euNzArHVqT7_UTIqBTJ28sFhi5w4rez13zdMfnQLktYTzQZUWBHg-
dQXHAe9bfMVG_qNwfh77ElECjv7a-8ntKm5kWEKZf9X7cJaPZ22jrVuLUl6IpeOsz1IWYu5KV-
FllDvvbep72YYlcVYBLPAaDfDlhWT3SNxKSUubJsh16XQh8xQIfmLmOmoyO6DmoJhNWOv2exoQ0m9Y4CrcrG3M1X3QFM6rA_fHrABfFlH145o7vjbW8ebf-KPNJUXbg-
SRIVd0lpsQQGEnjSC_Ld-VzwPJkiqyaYbelAb-
iCRhmISkeaj_9Px5p8O4tlf_tRV4Y9ubi3aJFKo59MZ1_QM730526PKPebBxLopSpd7IHGgvgtNSlVfbPZe7XYH3gjQmZIQ3WrBzy50KZuureD5Al7tWYQUffr_REEN2UAOaKbllroYSOT4rn
nbvCjN-ah9s7bK5vxjH3OmNakKTwA_E1Gqe1D6f3ZayUn43O68FSk_jzW6-
s2k6Zna6qhe610kMqgnapofw2ZX6hDgs0fw10kecmeh8lUiG7TJZSezviwfTNYcGZlIQz_xBgvZioUikHbNToZ7Rpey2DACOQp6wq-
L1CiwYnsaVCNkCDyYvHtnf4favu_Ys9AUNhlGFI98qsok1LjEgJj1d2Hf92VqkRqWdQvhTjSFeZ9gT76hAvXNwp3mWmdlouwbb0oBMzLWGpA79q3HhUveFoZuHcUt3rLteUETxPY7F0tuGfVn
W68kx7zSAdst_Vob40KmUOKfG0d9I19TtDgAvNuOZdTGVwZ9suCr3P_eiW2DYm6VqLkwmQqUX97W-mJokr9S8AhwjcDtBxXFJvkozEu5VwZ26dm-a3dfSxq9zJyF81-dwkeCs-
hdInJPxoHDHyub1tMsbxRylR2h-4iGtz1QuXk2Jn5JAfGYZKkqO5-
CQwZInWVMZBXI7GD7NTkaltYXNnVZna88WYUa7k1zM13PIJyRx8__VUwAmgBsfnZctU_4Kb1vVjZY9S8wfZmMAU8GDDoxyET5IwBZyu47bpljAwKrhaO0UVt9mTJBiiriVHBgR1PVqX_QZ57h
-nPJWQS5oGuxBexBlzz3SeK3y89OIdnzQcYpmvfutSxfI-QFCTJ5-
BJ6PRjcYi6F6zlz98lEjrz6J1YngIvDFuqb79L7NaxJOk2hsFBickJgtk-0He4X1cM5o20WFNbZKZyiItewn1ygn8toTiQfAmxs8gVGgi42SJnNPey_a4y9C84oJAuE10ojSYwi08jk6EdnPd
qU7C_9VnGHxXq6UVuI6wgO6ZQy71ZCTLFn_mf3Ap3rjAtt3TxZB7AEvBCH-jxOJXVFBOBdBBWqFcNM-swnZR5FGaWM2Vh-Vn-
zUeJ8yIU0343U7k_BC5_9P08rsUjblSVGbfaQXElxVZgy1d84caGDaaqB7ogogeAnjbnXwUF98TDMbgh57khXbb3romDL-
IfUbMRwiylpRIurBK6NXBmd03lPWXNsDjJTl1SsNlwuG6RCR9qtiF_6SOkxoYVEk9-cd0vrImTubrBBaNdpc-oCpF043fv65xnhSgaQH-
Vx1X7129B8o6BLVMyyh48prqlLUEE2vV3O0aIKCUhPw7iWqwzr0wPCodrJNxfnUjNN4pnYXpIszZNTl4jcnK6kKSc6sqlb7gAdPVMSh-
GWmEq9qbzuBGcMn0m7_2rGp0ud4VaLhFu7lfj8jITglklJFJ6QSd5ut6OXnJX4-
j8xLnuhUj2OdbQEeBCNJL9JNPZx0aWPRelDYQyh3iZ6aMrOE6eh6KCUYrCrbflL6zU0RABc9XDiI0Geeq2VlMNS4bhisFXm_3tPy78MW72bEB9xIBnYAhmy2w4l4y4bbo9G1HExCw8i8rn7t3
p05t6fPEqHwR1Fx-CxCE6Ilmxf_bgy0pPwSzprX2oSBlOjD2aSbDoKLk0NOYlBP_nh-
LCKBRujKmrHwJgUJoblh7msfZfg8QHDEqYkQoG3YADSHjaB3E5UNAvILJVlofpICFr6KR83MVM1kU7LUI2LBUkAiK2v8xp5yB-
P2UrnozksblE_xpUdsZqCJqayQ8cKEflPajLPN5GZORX8gdpelb3pnui4c1dx4BJdAKECUkr_Auv-hSKTtiHumo9q3T6f1Tyd-rgHJgU33n0cdvIN3S4asv2Glvwn1-
Ii5kYjiXHE8JDi2b6IY8NLoapCAkhI-3wfBPHIjl46YadtFojhPYBOsFNIOVrAc_7Yka3UYr2UAHaoVtXNglfpYQr6HmQ_tPPutsRRZbiC_QgIVgA4rjs_4Z3Nua4BTeCAX8ZEFos53DU3eIe
FNxgIcL4_RfKHS3T-rWPLz0MpEpxDOxGiTNQqUQc9XoW078iISELQiQjGrqOJtPswW6VpL833r7W_ac2xwWzbggvf7ms9zW84MA-
gkrdEENi4_X1Nc78szdAbW6dD60aFn3MEUZcTlD2o7M3NiqVKCcpjxIfXVOoX9MRamT71QDDRxyO2e5dbmmJjnMBWxjNt0ZQfhRr1lJstYld_YxbIBwf3cnLsTzmoGn6w9qDOAgzml43pjhuI
3jllSRb-nJSpm3VTyoyyxJbIEBQOXwLxNiOyRvCpnwyCLRyxKz5bPgh1DfaQ8PUK9K2dUq6314ynxDs9OEm_E1WgpBYI4Un8yTHFUUoICSHZqeb0QvH8MlHro-
Hpa8zyQsCUFGKcjOzAK2saH9e2twsml9nn-
iTaqs_ccaOcSxqd0mC4tUVEzz6fGlh8Epmv9U9Z69KAlphWBaDBPejSTaYnsT7m8hAPduwpNrFlrr_XWAoBghQRiF06SOwpAvauFuMYaE7LVRP6h4MxcmR4-9MWuE4ggiNxhEySb1rwefu1m2
hwhY3UmiY7pfGVBfUOxvhEIsUJkKTwimx0TPTVp1bo0uR7-
NuQlzyjIiXNG0j5uAtLP0uhofREaZrFoX0zEtG8u60TopHVIRn86GSKmEEwxacXcxyREfZaxWnD2tSqkuKeJW6RavwsPBAwSA2gWpqBIDpz6uzYIBJXwItqR9Bq-
hiix_KmpAg0CclvSQlxBjJeUFpe7YlYX8TddUwyQkCivwLVTvkvX7Tb5mGixZXIewlEKJL-znRpvHXK7L0M6xIb9XfWkF-0RhA5-k6YXyh4nnUrmF2sGgFFkKQZqzULY9VG2crjcynzu-
YCgasn3EvPo1oFWT1PMbyr5wUinEbBsaQ73cQ86HcNwtqP5Hf0tImXTo9FF60wwvvsfv-4arTTazPoD-2YshBzYRj0Di2BPpF5spfhtqYqnduqMglVa9btnxI9L1XUrJNeRrpokqMyN_io5ao
sOheF6KPWw_TtN5WAMYDJ-q6poujbSArWRYVnJ6YloAHD-38IXzSqp7aLGzO8qK12jiP0AW8Bfk3ZuzOK2x6NkawzG5yR7vadNu-r1eEFUMlo2cPlM98JzqpPqvMJ9R1_Z4w35H0qUXeY8-
Fo6PTp4gbvcAlxnETcgqjzPy75TczHrG6Mnym2tFP_4R5n-tcBm-5bS5ZsVRe-
dKkkkP3TtjNH1NSbrx1OchkW_cQd_37uhMLI1kSzVZ_QutWSBdcKL2mJgtzeVHwH6ADEp4TaMlKy9JP5PNrCrkHlyX5njTxrMoHyUNa34MgH_3r6phq8D3zT6Jz2dMuE9pjHgXpa2aSYFMoVj
1hmu-i8ul8a0OImDsQN-aTVHMtyF-ojj0Bj0vX6axmAR6G9n39VIh3lTd7KqDjCGckbEHrUvrKRCcs0F9HfOnPv54v1IMJkUS1GW4-
abSRyteNfCqrPsGr7amCKwjLQmbdqeCUPTHrAfIWfiMJ2fXyAVLDL1NNRja6BKq_3b4F364Y14mxBValIe5pwmP0sCfnbfglcapq2V6qcyFwxzqM4gy9DrDg0z-
y0Fe6Mbax3ZPO1S712I8gNHFn19airb7nE6AsM0WkRPu6L3nuHffFrTvE9hJQrlqlARrL4zCYTtvZPFQbjrdTb4bWXjztO-
Nnn885ZFnbdDeZgCiVJAv3ySyvMvPxBylnTJRsMqht553JCV0ULK5SOn1MnLGL8QAEi4f4R23cj-
tsjB6QMLFeg5cFdBfZ7YiJ51objvowVKZJ5hYnitbz-5ZrRtnnD9gY50VzRGykfmYXAx9CzvNfyQMj7KeiaBVTdPN08qYimQjIquSFw0j3b3WlyqXB_ILMqYh3Z13jYjo3YgzYC57l5BNI7zs
GaBafgwpX0nRADNX-WUlVJqhPg4mOSY7rWn1Q_UFWYAhc_Ok8unkMtpngbug556m_wfYIi5Q-
i2nsAjPLA3btP0W4mKs1aqdpmYGTEefSXNHxkeqXESTOVsY0RP12sJIfC5lBvnNkiT6R8k6FdVX3syHss-
IiUDQVjh_medNsI4LNTE1I5kYrTIuJnt0EJmzhs113TKxPb5fKvY15JZnjRYI4lcHDa1CJ80Bkt-
MKFlmqfnl64CdpiiIspcet3MPuIdvvnLCKHYmaj9q1lVS4W8Hr7yS76I6Vjifu2BKlmVwVakqhP7cPKijdNjjmsIYWdIMQ72Y7nX7bSDkORqQQ0frr1wYkg5MahHbAKmv42ISb8ARFNXJq0qC
Fq3ZC09MqMNwfYSShnJSAW8LvAmVGxptwi4j9S2yG3ERQnRiVou5znhp4BM7pW8hs5cyIe8ZVUhymJiHFwtZ6lS3TgT4XuJ7nM32U6vX3G3tEhbYfQuU2W8cMws_gG-pHstJimeLWALhf4M-
vPeNxaMpC3hlIhyWPXfX1Bm-1gVUtD3Vu6ujthpAWuy5LbNeEdVC3QJzu6T_BFPP2E28aFoLSGWI6tjXGXzbb4tA8U45QH7Idpl62FbZf3JEoRMEScOfXPIH1RxT97P-
n4U2xnkwe6Z9xs64pmBYZ_HJcMFujglk9pJ4kDQm2DuIQwCGCpo6q9aNAsY2Ibvke2HRpuklFUf04jIZ0ogbTaTTt19Rsnmac_0NsaoeCssSrQtcoL5aPUk1Xix3tLuB5aXdoNjGEbmmsY6hN
8nek2aaK_2hQUGgy2HvNDTkuPXR20NSoVsOo9iqZbeiiU624FUPtPj7Tn21dLfqRdMQ8MXmN1uWmKl2YiiefqviYzPgNyZJvejNrlmK_ImFH7zmO5NTjb88G2Pt95tXYeU5crYYaoVaW6GaNW
tTyHJQ287XuCFiBGjqy3nWvl8cHtVthDriUA4rhV_lGPlJfpPY-pcWxiJ7KKEj0scfEFrniIjXUWXc0e2bxjUBNmICJrR9ElfIuOD8wc6hYt9E3lKy4-
l44Bs4II3IPd9n5pxVOGqP8wdVGgaf6SQLuy915JUXEoRIWlCePm2fYNyoNJGriPFWPmaBRWbKHDLTFTfrhWbal8sYJCePMllipsUgSldNbTuhkQxU1hZ4fvW-1HAnfLNyGREmHNDmYNPFwDW
d6CR6exKjNSomrvRvfxpW4bfCt5Ttsr1GgGwnLCvoyq_mkXIYNejSs8gIUEQMqY8ZkOwWupoXTHXzJS3hEreJqNQ0NCW4QvNXBoDitQlHFpeh4z1xUpDewmVgeQkLjSq-5Mm574ViSQHJcMDp
isSd9aaL2OxweJIrFWEQBvRgDs0EPZ--kSyMEDcFeNYuYi6XTk8hzQk67DeWIeXCDLFlk5BlNK5kORNDVOGY_Lka-dIM1srXxqaUiZYBwncvbvRUuhciPLxjKV-idWrxELHbr0Hmeo-
sKIw6YfsdlwZrJ9uSe5fc9pEWXTbu2hkuTFG8_Wxi0lOnFws_AmDydmT32W5NQso3vDm_NXASBrRE","tag":"ZVFLPgLaA8wdtZPgfVoZ8w=="}HTTP/1.1 200 OK
Content-Length: 0
Content-Type: application/octet-stream
Date: Tue, 12 Jan 2021 18:47:53 GMT
Server: Python/3.6 aiohttp/3.5.4
```

Figure 4.6: Traffic through the DIDComm protocol is encrypted [68].

A potential threat to this infrastructure is the possibility of the participants' computational devices becoming compromised. In that situation, a malicious entity would be able to act as their victim and participate unauthorised on the ML training or issue illegitimate VC to other unauthorised partipants in case the compromised participant is the NHS Trust or the governmental regulatory authority. This threat is common in infrastructures that utilise identity certificates to interact; hence, other security systems should be in place to mitigate this threat, such as Two-Factor Authentication (2FA) techniques [328]. Another threat to the presented infrastructure is the possibility of DDoS attacks [329], and in order to defend against it timeout countermeasures can be set to block enormously large traffic deriving from one of the participants. Another countermeasure against DDoS, could be to setup the infrastructure in a Kubernetes cluster to handle the load balancing and recovery from failures [330]. Finally, common cybersecurity countermeasures could be set in place to protect the infrastructure against cyber attacks. OWASP is an open-source organisation that provides the guidelines and techniques that could be used in order to mitigate a number of cyber threats [331].

4.3.3 Summary

In this section, TFL presented, which is a combination of SSI and FL [68, 47]. This is the first proof-of-concept that merges these technologies since the original scope was related to the exchange of basic text messages. Additionally, insightful metrics about the infrastructure's performance and security were provided in Section 4.3.2. More specifically, the scope of the presented experiment was to create a trusted healthcare ecosystem in which three hospitals can train a ML model using their sensitive data (a mental health dataset) securely and privately without sending raw data directly to the ML researcher. Within this trusted ecosystem, two regulatory authorities supervise the communications, the governmental regulatory authority and the NHS Trust, which issue the legitimate participating credentials for the researcher and the hospitals. The communication between the key participants occurs only if the researcher and the hospitals hold legitimate VCs issued by the regulatory authorities that are able to successfully resolve the public DIDs stored in the public identities blockchain ledger. Additionally, all the communications and the transmission of the ML model between the participants occur through the end-to-end encrypted DIDComm protocol, which prevents unauthorised intermediaries from intersecting the communication [68].

As presented previously, TFL provides a trusted ecosystem to conduct FL more securely; however, common threats against FL itself, such as model poisoning, model inversion, membership inference and adversarial examples, are not mitigated by default.

Hence, as a future avenue, the extension of this architecture by combining it with other privacy-preserving machine learning techniques such as DP [332, 233, 333], SMPC [334] and adversarial training [69] is crucial.

4.4 Conclusion

This chapter is focused on the trust establishment through decentralised identities and the various ML approaches that can be implemented through end-to-end encrypted communication channels. A novel ecosystem is presented that facilitates trusted FL between three hospitals with sensitive data and a researcher with a ML model. The ML model is being distributed to the hospitals and trained on their private data instead of circulating raw, sensitive data in a central location. All the communications between the participants occur within the end-to-end encrypted DIDComm channels that have been experimentally evaluated [68]. Furthermore, the system presented in this chapter can be used in combination with other privacy-respecting systems (Appendix D), such as the PyDentity [335] and PyVertical [336].

Future avenues of the presented experiment would be its combination with distributed ledger systems, as seen in the previous chapter, to create a unified system that will preserve the privacy, security and trust of the used ML approaches. However, as presented in the literature review, a set of ML attacks still cannot be mitigated fully even if trust is established between the participants, despite all the efforts and privacy-preserving ML techniques. Hence, the impact of some of these attacks alongside further countermeasures is presented in the next chapter.

Fifth Chapter

Adversarial Machine Learning

5.1 Introduction

This chapter addresses the **Objective IV** by presenting a practical experimentation using adversarial ML techniques in an IDS environment, experimentally evaluating their impact and presenting suitable future avenues for mitigation techniques [69].

Adversarial ML is a crucial topic, and even its popularity, especially in the last few years by the research community, there is still a gap regarding the real-world impact of these attacks, as well as thorough mitigation techniques against them. Additionally, as the number of methods, techniques, systems and technologies increases, the attackers' potential attacking surfaces also increase [69].

Systems such as IoT devices are a common target to malicious adversaries. The reason for that is due to the fact that IoT devices are commonly sensors and other low-specification devices without adequate computational resources to perform their operations as intended and also to be configured to preserve the privacy and security of the infrastructures. Often in these devices, security is an afterthought, with a few cybersecurity attacks originating from IoT devices such as the Mirai botnet [151]. Further cybersecurity countermeasures can be taken to protect these vulnerable devices, such as the usage of IDS/IDPS. IDS aided by ML approaches have been prominent in recent years since they are able to detect threats more efficiently even in situations where the threats are zero-day [337].

However, despite the advantages of ML IDS, there are also potentially threats associated with them. One of the terrifying threats against these systems is the possibility of an adversary evading detection by disguising the attacking strategy comparable to a genuine approach. This type of attack, specifically against ML algorithms, is called adversarial examples, which aim to trick the ML systems into classifying malicious activity as legitimate. Currently, it is very challenging for the ML systems to mitigate this type of attack completely, despite efforts against it seen in the literature [338].

5.2 Adversarial Attacks against Network Intrusion Detection Systems for IoT

ML-based IDS have been increasing in popularity in recent years. However, adversaries may use an evasion technique, namely adversarial examples, to circumvent detection [176, 65, 177, 178]. In order to make the ML-based IDS more robust, an adversarial training technique developed with promising results [65, 179].

Additionally, the popularity of IoT devices increases as well and often, physical objects have the ability to connect to the internet [31]. However, as mentioned previously, adversaries aim to exploit these devices that created without security in mind for their benefit [151].

This section presents a promising study that evaluates the Bot-IoT dataset [169] in adversarial examples settings, such as generation and label noise attacks [69]. This is the first work of its kind. Firstly, an SVM model was developed for the classification of the records in similar settings to the work of Koroniotis et al. [169]. Secondly, an ANN was developed and trained using the Bot-IoT dataset with similar activation functions as the RNN and LSTM in the work of [169]. Finally, the scope of this work differentiates significantly and extend the work of [169], towards the generation of adversarial examples for both models and the experimental comparison and evaluation of their impact.

5.2.1 Architecture and Methodology

Prior work investigated thoroughly the impact of adversarial examples using the KDD-CUP-99 and NSL-KDD datasets [339]. However, Bot-IoT is still recent and unique in terms of accumulated data records. The authors simulated an attacking-victim experimental environment, in which they captured the attacking traffic in *.pcap* files and exported them to Comma-Separate Value (*.CSV*) files [169]. The results of their experiment was the creation of the Bot-IoT dataset which consists of 73 million data records, 46 various features and 3 features used for classification purposes. As it can be seen in Table 5.1, the attacks have been categorised by the authors into: i) Normal traffic, ii) Reconnaissance traffic, iii) DDoS traffic, iv) Dos traffic, v) traffic related to Information Theft. Since the importance of the feature selection is immense, a list of top 10 features extracted from the dataset, with the results presented in Table 5.2 [169]. Additionally, the other features present the three classification features and the values they can get. The first classification feature is *attack* with the classification labels *true* or *false*. The second classification feature, namely *category*, takes one classification label from those seen in Table 5.1. Finally, the third classification, namely *subcategory*, includes more specific details than the *caregory* feature, such as including the specific transmission protocol (HTTP, TCP, UDP) in the DoS category.

Table 5.1: Bot-IoT category value counts [69, 169].

Category	**Full amount**	**5% amount**	**Training amount**	**Testing amount**
DDoS	38,532,480	1,926,624	1,541,315	385,309
DoS	33,005,194	1,650,260	1,320,148	330,112
Normal	9,543	477	370	107
Reconnaissance	1,821,639	91,082	72,919	18,163
Theft	1,587	79	370	14
Total	73,370,443	3,668,522	2,934,817	733,705

Table 5.2: Total features in the Training dataset [69, 169].

Features	Description
pkSeqID	Row Identifier
Proto	Textual representation of transaction protocols present in network flow
saddr	Source IP address
sport	Source port number
daddr	Destination IP address
dport	Destination port number
attack	Class label: 0 for Normal traffic, 1 for Attack Traffic
category	Traffic category
subcategory	Traffic subcategory

Top-10 Features	Description
seq	Argus sequence number
stddev	Standard deviation of aggregated records
N_IN_Conn_P_SrcIP	Number of inbound connections per source IP.
min	Minimum duration of aggregated records
state_number	Numerical representation of transaction state
mean	Average duration of aggregated records
N_IN_Conn_P_DstIP	Number of inbound connections per destination IP.
drate	Destination-to-source packets per second
srate	Source-to-destination packets per second
max	Maximum duration of aggregated records

5.2.1.1 Attacks Composition

The developed SVM model follows a similar setting to the work of [169]. Hence, similar outcomes were produced, such as the SVM model's accuracy, recall, precision, and F1-score. The developed ANN is close to the RNN and LSTM models presented in the work of [169] for an unbiased comparison. For the creation of the adversarial examples, the labelled noise generation technique followed during the ML training, which performs better in traditional ML models such as SVM, instead of feature noise generation.

For the first experimental test, three different label manipulation methods were performed, and their results were compared and presented in the following subsections in terms of the set ML outcomes. Firstly, a percentage of the labels flipped ranging from 0% to 50%, followed by random label flipping, and finally, targeted label flipping. For the target label flipping, the features with the greater weight were flipped; hence, they had the most significant impact on the ML model's accuracy.

For the second experimental test, FGSM approaches were performed, which consider that the adversary is able to alter the data used to train the SVM and ANN models. In this scenario, a potentially adversary data provider would aim to trick the ML models after their deployment into classifying their inputs wrongly. To simulate the attacking environment after the ML model's deployment, the CleverHans framework [340] was utilised. The extend of the discrepancy between the manipulated and non-manipulated labels is defined by the noise factor. In the work of Wang [341], the authors revealed that a noise factor of 0.02 managed to return 100% inaccurate classification utilising the NSL-KDD dataset; hence, in the presented experiments, the noise factor increased in increments of 0.1 to a range from 0 to 1 since the chosen Bot-IoT dataset is significantly larger than the NSL-KDD dataset.

A matter of great importance in the presented experiments is the evaluation of the findings. Various ML metrics, such as the accuracy, recall, precision, and F1-score of both ML models alongside their confusion matrices, were measured and presented. Particular interest was given to the false-negative classifications since, in the settings of these experiments, they mean that an adversary could potentially evade the detection of the IDS and launch their attacks.

5.2.2 Experimental Implementation

5.2.2.1 Data Preparation

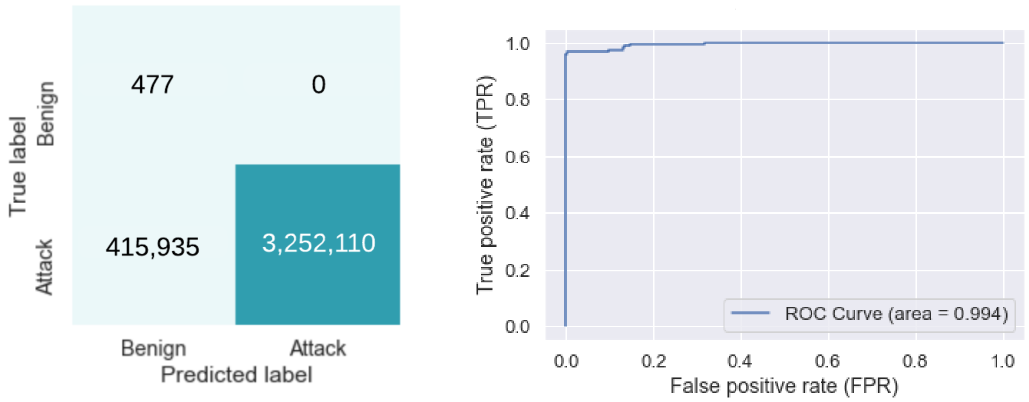
The first step of the experiments involves data preparation. As seen in the work of Koroniotis et al. [169], the authors utilised the Correlation Coefficient and the Joint

Entropy Scores in order to identify the top-10 features of the Bot-IoT dataset, as seen in Table 5.2; however, as seen in the same table, nine more features added due to their usefulness. Additionally, to simplify the ML procedure, the authors extracted 5% of the total number of records corresponding to 3,6 million records. This “smaller” dataset was used in the presented experiments for more manageable control. Finally, the authors split the records to 80% training and 20% test sets; hence, the similar split sets used in the presented experiments to keep the architecture as comparable as possible. The training and testing data were already in numerical values; hence, encoding them is not required. However, normalisation is still needed to scale the training and testing data values contextual ranges to the range of -1 to 1.

Finally, to maintain the integrity of the experiments, a subset of the dataset marked as *trusted* and specified the data that have not been under manipulation. This trusted dataset would be compared with the manipulated dataset to investigate the impact of adversarial examples.

5.2.2.2 SVM Model

The developed SVM model is similar to the work of [169], with specific hyperparameters set in similar values, such as the penalty score set to 1 and the maximum iterations to 100,000, as well as four-fold cross-validation. As it can be seen in the confusion matrix in Figure 5.1a, 415,935 attacks predicted as benign traffic. Further, the SVM model’s ROC curve can be seen in Figure 5.1b; hence, the model’s accuracy is 86% to predict an attack successfully, as it can be seen in Table 5.3, alongside the rest of the ML metrics, such as recall, precision, and F1-score.



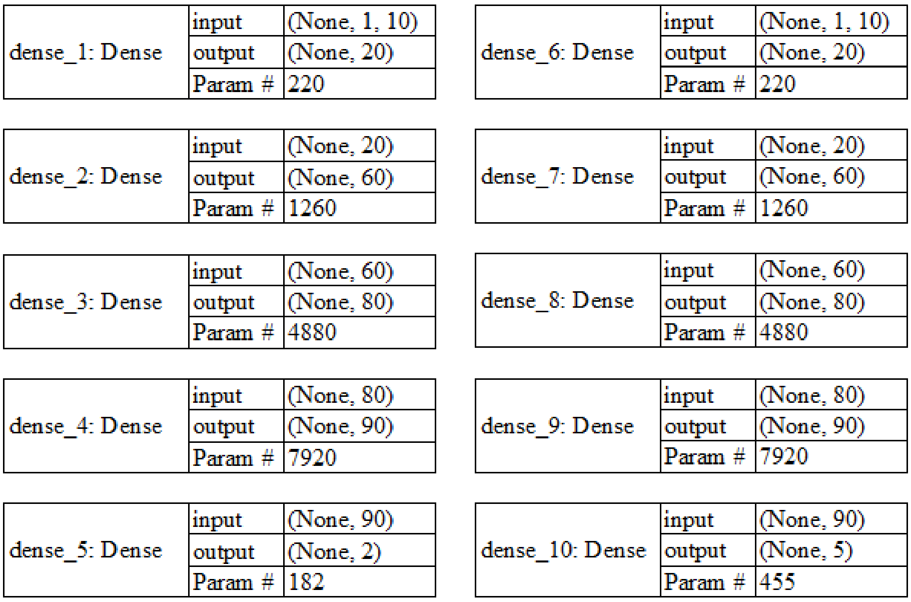
(**a**) (**b**)

Figure 5.1: (**a**) Support Vector Machines confusion matrix without label flipping. (**b**) Support Vector Machines Receiving Operator Characteristic curve without label flipping [69].

5.2.2.3 ANN Model

The developed ANN's structure can be seen in Figure 5.2 and is composed of one input layer, one output layer, and three intermittent layers. The left side of the figure displays its structure for the binary classification, whereas the right side displays its structure for the multi-class classification. The input layer of the ANN is composed of ten nodes comparable to the number of ML features. The output layer is 2 for binary classification or 5 for five-class classification. Additionally, the Sigmoid activation function is used similarly to the work of [169], which also has been identified as more robust against adversarial examples [173]. The intermittent layers are 20, 60, 80, 90, respectively.

As it can be seen in the confusion matrix of the ANN's binary classification in Figure 5.3a, 2955 attacks were predicted as benign. Additionally, regarding the five-class classification, as seen in Figure 5.3b, the ANN predicted more labels as Dos and DDoS traffic instead of their legitimate labels. Finally, as seen in Table 5.3, the ANN resulted in very high ML metrics, while its loss remained low.



dense_1: Dense	input	(None, 1, 10)
	output	(None, 20)
	Param #	220

dense_6: Dense	input	(None, 1, 10)
	output	(None, 20)
	Param #	220

dense_2: Dense	input	(None, 20)
	output	(None, 60)
	Param #	1260

dense_7: Dense	input	(None, 20)
	output	(None, 60)
	Param #	1260

dense_3: Dense	input	(None, 60)
	output	(None, 80)
	Param #	4880

dense_8: Dense	input	(None, 60)
	output	(None, 80)
	Param #	4880

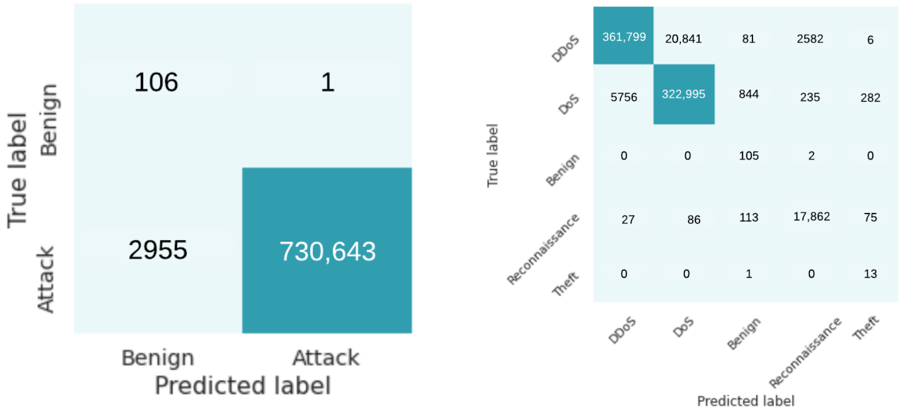
dense_4: Dense	input	(None, 80)
	output	(None, 90)
	Param #	7920

dense_9: Dense	input	(None, 80)
	output	(None, 90)
	Param #	7920

dense_5: Dense	input	(None, 90)
	output	(None, 2)
	Param #	182

dense_10: Dense	input	(None, 90)
	output	(None, 5)
	Param #	455

Figure 5.2: Artificial Neural Networks Design [69].



(**a**)

(**b**)

Figure 5.3: (**a**) Binary Artificial Neural Networks confusion matrix without adversarial examples. (**b**) Multi-class Artificial Neural Networks confusion matrix without adversarial examples [69].

Table 5.3: SVM and ANN scores without label flipping and adversarial examples [69].

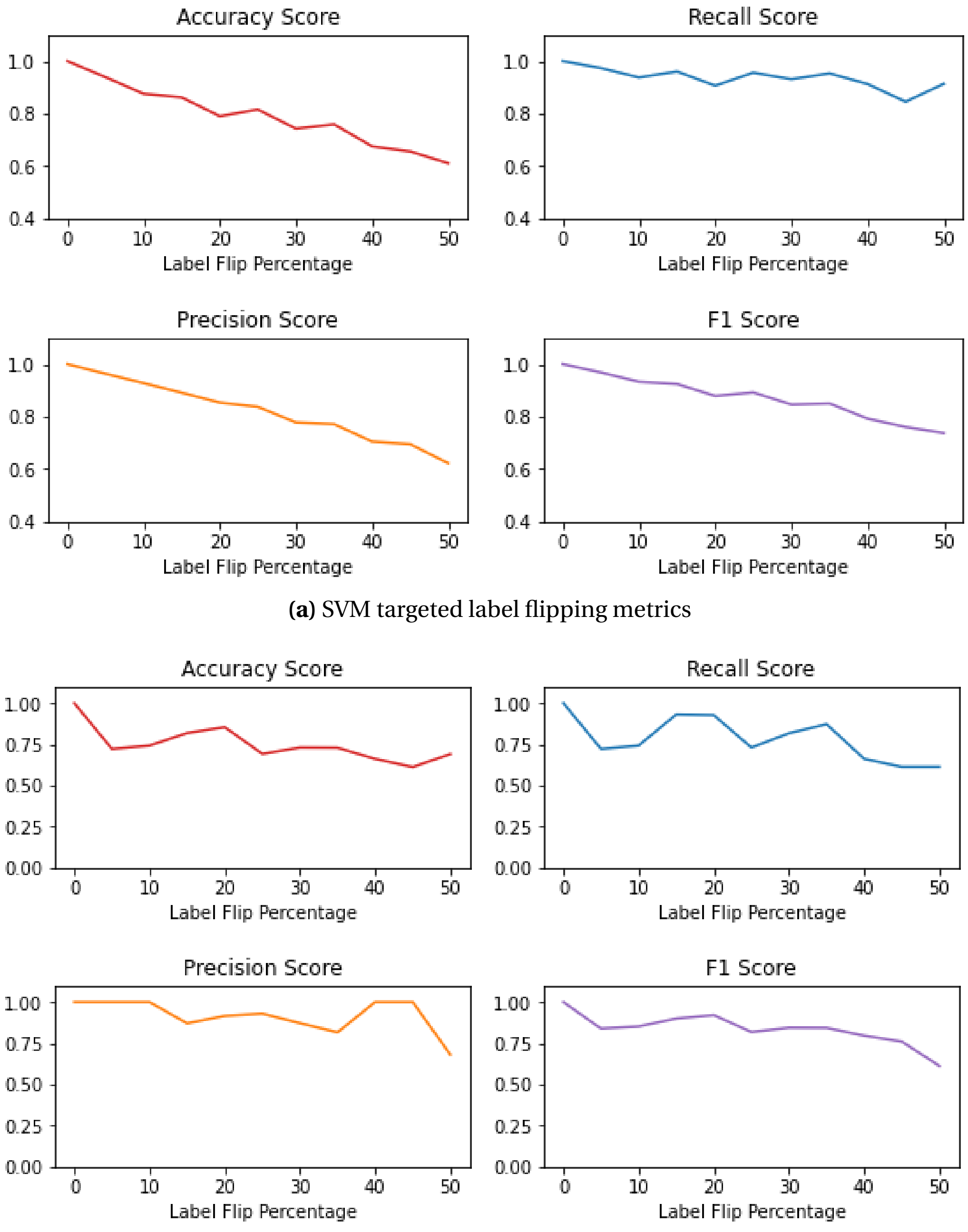
(a) SVM scores without label flipping	
Scoring	**Percentage (%)**
Accuracy	85.897
Recall	85.895
Precision	100
F1	91.255
(b) ANN scores without adversarial examples	
Scoring	**Percentage (%)**
Accuracy	99.692
Loss	1.170
Recall	99.813
Precision	99.591
F1	99.702

5.2.2.4 Generating Adversarial Examples

The experimental activities regarding the generation of adversarial examples are divided into two subcategories. Firstly, regarding the SVM model, random and targeted label flipping activities were performed to the training dataset. Secondly, regarding the ANN model, the CleverHans library utilised to generate the adversarial examples [340]. This library also offers random and targeted FGSM approaches, with the random FGSM aiming to modify the labels on the dataset, whereas the FGSM targeted approach aims to alter the dataset as much as needed to maintain it similar to the classification feature.

SVM Model. Regarding the SVM model's random label flipping, the labels flipped in ranges from 5% to 50%, in 5% increments. The *attack* column is flipped from 0 to 1 and vice versa in the training set. Regarding the SVM model's targetted label flipping, the labels with the greatest importance should be identified utilising the SVM hyperplane. The labels with the lowest distance to the hyperplane (margin) have the most importance; hence, their labels should be flipped from 0 to 1 and vice versa to generate sufficient targeted adversarial examples. After the label flipping, target or not, the same data preparation techniques are performed, as seen in the previous subsection.

As seen in Figure 5.4a, in the SVM model's targeted label flipping activities, as the range of flipped labels with high importance increases towards 50%, all the ML metrics are being negatively impacted; hence, the SVM model is manipulated successfully. The non-targeted label flipping activities flipped the labels randomly without considering their importance; hence, as seen in Figure 5.4b, there is no direct correlation of the increasing percentage of the flipped labels to the ML metrics. The specific results from the presented experiments can be seen in Table 5.4.



(a) SVM targeted label flipping metrics

(b) SVM non-targeted label flipping metrics

Figure 5.4: SVM model's metrics during label flipping activities [69].

Table 5.4: Effect of Zero vs. 50% label flips against the metrics using hyperplane margin method [69].

Scoring	Accuracy		Precision		Recall		F1	
Percentage of flipped labels (%)	0	50	0	50	0	50	0	50
Random Flip	0.999	0.441	0.999	0.610	1.0	0.613	0.999	0.612
Targeted Flip	0.999	0.610	0.999	0.621	1.0	0.913	0.999	0.737

ANN Model. Regarding the ANN model's label flipping, the adversarial examples generation activities are similar for the binary and the multi-class classification. As mentioned previously, the CleverHans v3.0.1 library is used to generate adversarial examples [340]. Firstly, the Keras model needed to be converted to a logistic regression model. Secondly, the training dataset needed to be converted to a Tensorflow tensor. Utilising the CleverHans library, the value of epsilon could be set that defines the level of perturbation, as well as if the FGSM is a random or targetted attack. The generated dataset could be then used as a testing dataset to evaluate the ANN model, identify the ML metrics and construct its confusion matrix.

As seen in Figure 5.5a, the targeted label flipping activities using the FGSM for the binary classification ANN model, shown that the ML metrics reduced (apart from the loss that increases) as the epsilon value was increasing. This finding was expected since a greater value of epsilon impacts the data more. Additionally, after the epsilon value of 0.5, the ML metrics are impacted the most. Similarly, as seen in Figure 5.5b, the ML metrics also reduced regarding the non-targeted label flipping activities using FGSM for the binary classification ANN model. However, for this experiment, increasing the epsilon value over 0.2 started affecting the ANN, with the greatest impact after the epsilon value of 0.5, as previously. The specific results from the presented experiments can be seen in Table 5.5.

Regarding the multi-class classification of the ANN model, as seen in Figure 5.6a, the targeted label flipping activities have shown that the ML metrics gradually decrease analogous to the increase of the epsilon value until the maximum. However, in similar non-targeted label flipping activities, the ML metrics abruptly decrease after the epsilon value of 0.1.

In Figure 5.7, the confusion matrices of the experiments can be seen. In Figure 5.7a, the binary classification confusion matrix presented for the non-targeted and targeted label flipping activities respectively. The non-targeted binary classification ANN incorrectly classified as benign a high number of attacking samples, resulting of a recall increase in Figure 5.5b. In Figure 5.7b, the confusion matrices of the five-class classification ANN can be seen, observing that with a high value of epsilon, the ANN does not classify DoS and DDoS attacks correctly, whilst incorrectly also predicting benign, reconnaissance and theft attacks.

Table 5.5: Effect of Zero epsilon vs. 1.0 epsilon against the metrics using Fast Gradient Sign Method in targeted and non-targeted modes [69].

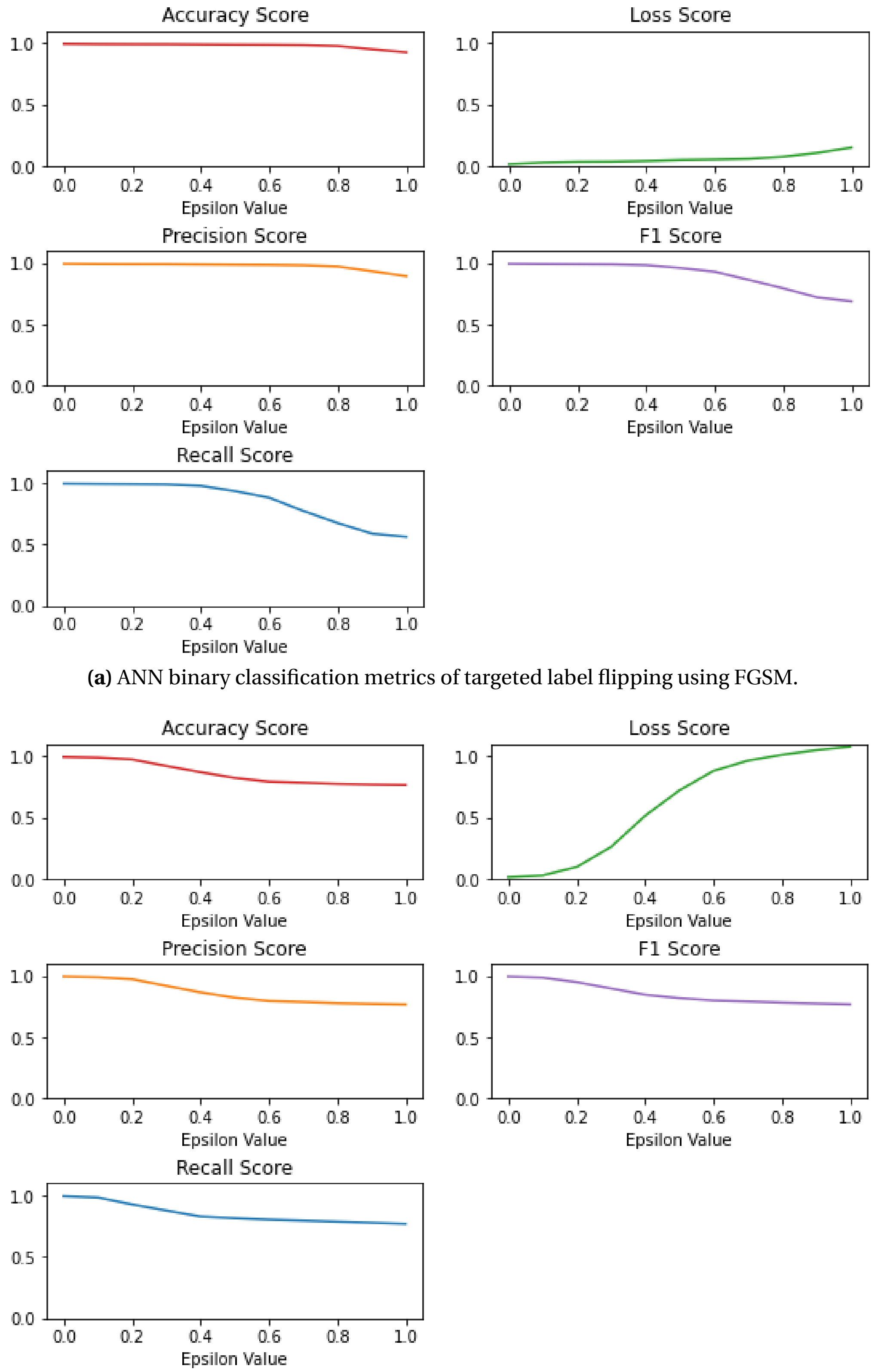
Scoring	**Accuracy**		**Loss**		**Precision**		**Recall**		**F1**	
Epsilon	0	1	0	1	0	1	0	1	0	1
Binary Tar.	0.996	0.927	0.016	0.151	0.996	0.895	0.996	0.563	0.996	0.690
Binary Non-Tar.	0.996	0.768	0.016	1.080	0.996	0.769	0.996	0.771	0.996	0.769
Multi-Tar.	0.956	0.421	0.045	1.764	0.952	0.312	0.957	0.493	0.955	0.382
Multi-Not-Tar.	0.956	0.141	0.045	2.403	0.952	0.153	0.957	0.249	0.955	0.189

5.2.3 Experimental Evaluation

For the experimental evaluation, a set of ML metrics was firstly measured and compared to identify the usefulness of the models. Further, since the experiments are within the IDS domain, particular focus is given to maintaining a high accuracy score with a low false-positive rate [342], which would be malicious threats bypassing the security.

5.2.3.1 Evaluating the SVM model

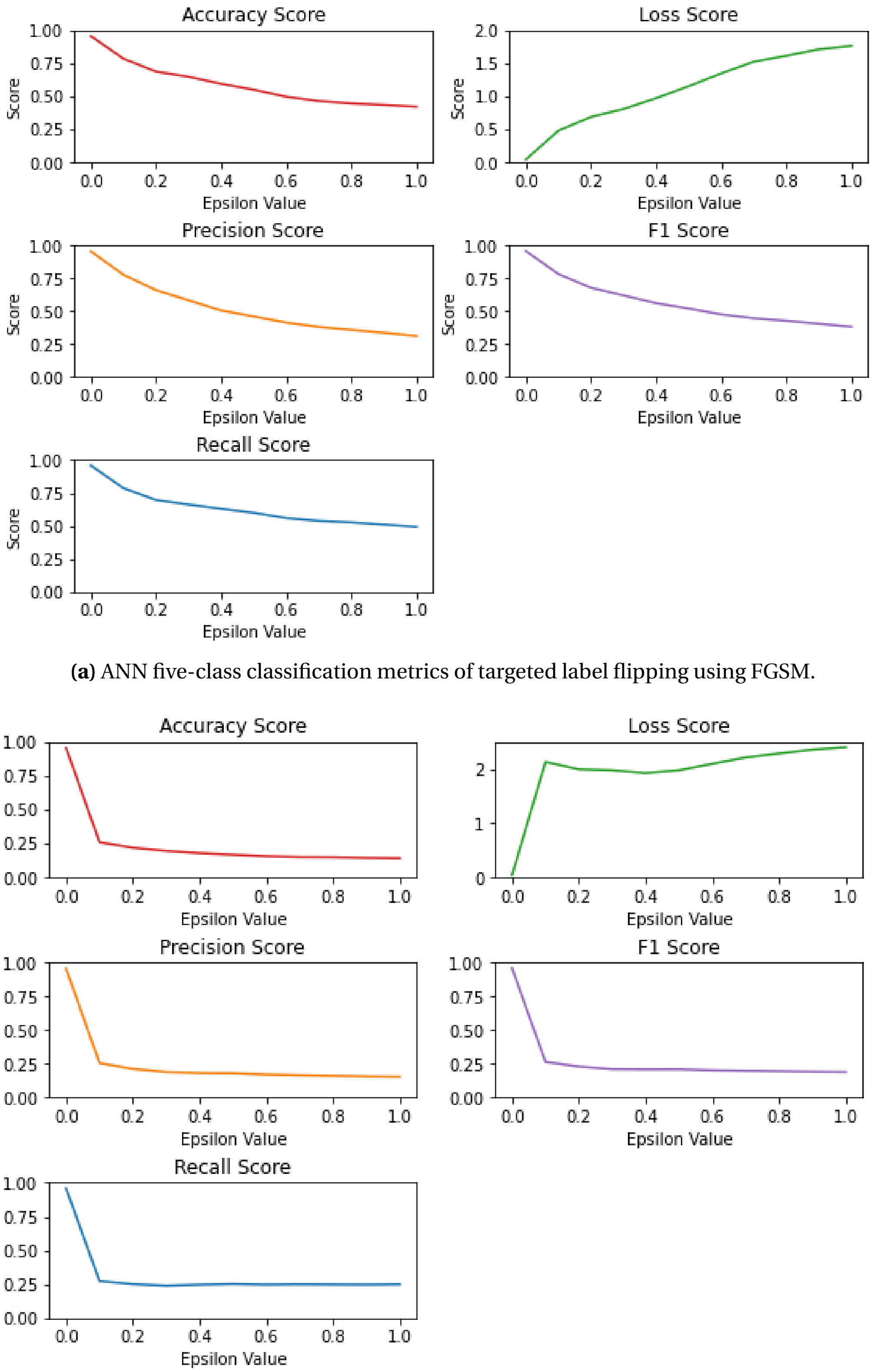
As the false-positive rate is increasing the recall score of the SVM model decreases and vice versa. The recall metrics for the SVM model can be seen in Figure 5.4 and Figure 5.8. By observing the Figure 5.4a, it can be seen that in targeted label flipping, the accuracy of the SVM model decreases quicker than its recall. That is because the false-negative rate does not increase as fast as the false-positive rate. However, this is not the case for



(a) ANN binary classification metrics of targeted label flipping using FGSM.

(b) ANN binary classification metrics of non-targeted label flipping using FGSM.

Figure 5.5: ANN binary classification model's metrics during label flipping activities using FGSM [69].

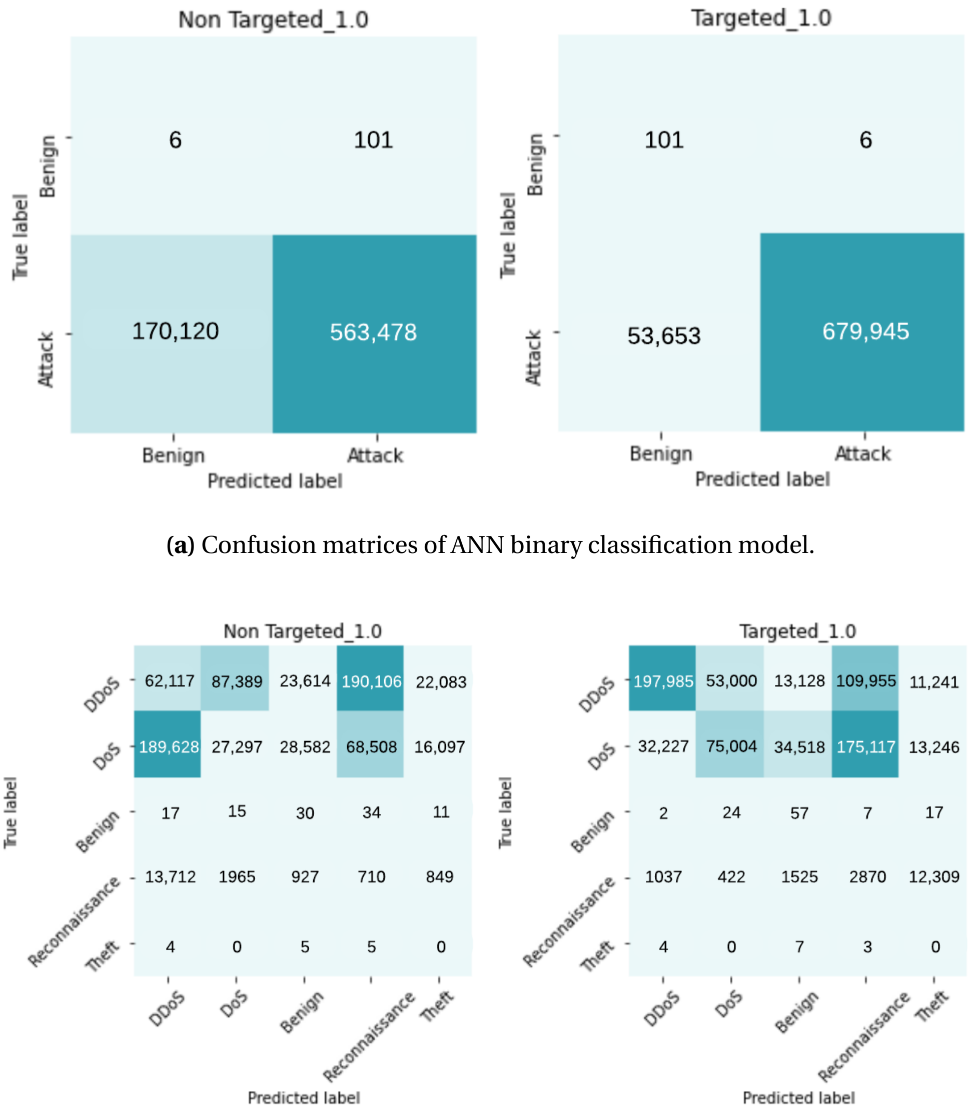


(a) ANN five-class classification metrics of targeted label flipping using FGSM.

(b) ANN five-class classification metrics of non-targeted label flipping using FGSM.

Figure 5.6: ANN five-class classification model's metrics during label flipping activities using FGSM [69].

the non-targeted label flipping, as seen in Figure 5.4b. In this experiment, the recall score is in pair with the SVM model's accuracy, which means that the false-negative rate



(a) Confusion matrices of ANN binary classification model.

(b) Confusion matrices of multi-class perturbed with 1.0 epsilon.

Figure 5.7: Confusion matrices of binary and five-class classification ANN models using FGSM perturbed with 1.0 epsilon [69].

increases quicker than in the targeted experiment. Fundamentally, that means that an adversary performing random label flipping would be more prone to exploit the ML IDS, avoid its detection and deploy their attacks instead of targeted label flipping. When comparing these findings with those from the literature, it can be noticed that flipping labels with a large distance from the hyperplane is required to manipulate the model more effectively; hence, choosing labels with the largest distance from the hyperplane instead of those with the lowest would affect the SVM model more, and make it more prone to adversarial examples.

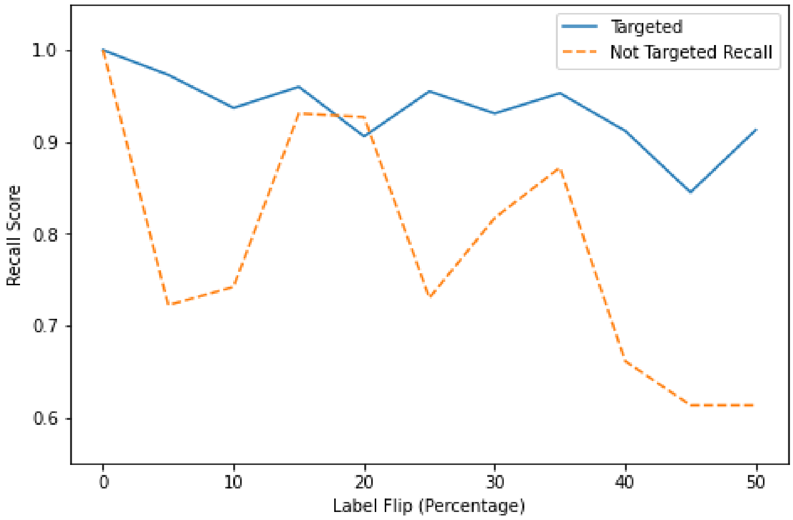


Figure 5.8: Comparing recall scores in targeted and non-targeted SVM label flipping [69].

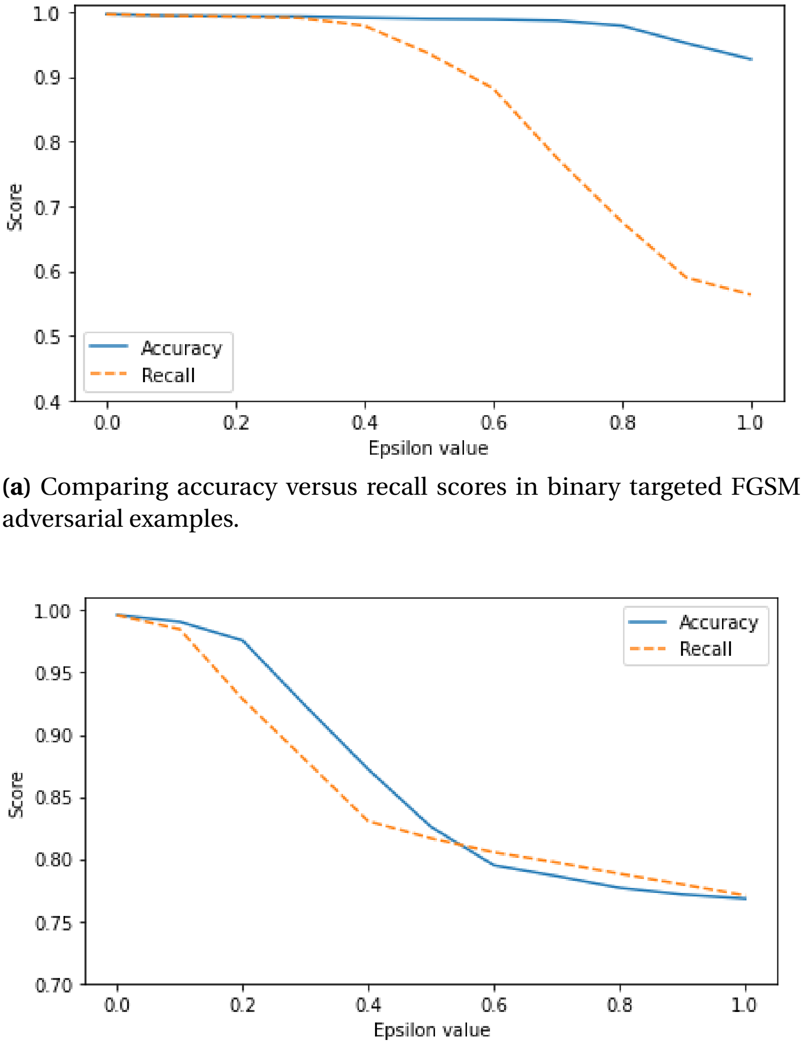
Another comparison with the findings from the literature is related to the experiments' threat model and its impact on the final results. In the presented experiments, it was assumed that the adversary is able to access data used for ML training. In the work of Huang et al. [180], the authors mentioned that adversaries able to manipulate data used for ML training would greatly impact the ML metrics negatively. However, as can be seen in Figure 5.4 and Table 5.4, a high amount of perturbed labels is required to impact the ML metrics significantly; hence, by using defences such as statistical analysis, these attacks could be easily detected [179].

5.2.3.2 *Evaluating the ANN model*

As seen in Section 5.2.2, the developed ANN is generating adversarial examples using FGSM and is split into two submodels, one for binary classification and one for a five-class classification. The threat model for these two submodels remained the same; the adversary is assumed to be able to manipulate data used for ML training and also trying to evade detection from the ML IDS after its deployment. As it can be seen in Figure 5.5 and Figure 5.6, the generation of adversarial examples negatively impacts the measured ML metrics; hence, the ML IDS is not able to effectively detect the threats. Adversaries following this threat model would be able to exploit the ML IDS and launch their attacks effectively.

Regarding the binary classification submodel, the generation of adversarial examples in targeted FGSM settings revealed that after the epsilon value of 0.8, the accuracy of the ANN model decreased. As mentioned previously, the increase of the false-negative rate allows the adversary to remain undetected from the IDS. As seen in Figure 5.9a, when the epsilon value reaches its maximum value of 1.0, the accuracy drops from to 92.7% from 99.8%, whilst the recall of the model drops to 56.3% from 99.7%. Additionally, in the Figure 5.7a, the confusion matrix of the FGSM adversarial examples' manipulation revealed a substantial increase in false-negatives than the similar confusion matrix before any manipulation, as seen in Figure 5.3a. The key finding from these targeted experiments is that even if the model's accuracy remained high for the classification task, its recall decreased dramatically, and means that if an organisation considers only the model's accuracy may fall victim to undetected FGSM adversarial examples. As seen in Figure 5.9b, the non-targeted FGSM adversarial examples have a greater negative impact on the accuracy of the model since it drops to 76.8% from 99.6%, whilst its recall score drops to 77.1% from 99.6%. The accuracy drop is because non-targeted FGSM aims to manipulate the labels in such a way, making them incorrect for the classification task. Finally, considering the confusion matrices of these experiments in Figure 5.7a, it is observed that the number of false-negatives increased in non-targeted FGSM settings. Additionally, as seen in Figure 5.10a the recall score of the model is exceptionally negatively impacted when using targeted FGSM adversarial examples since the accuracy of the model remained very high.

Regarding the five-class classification submodel, the generation of adversarial examples using targeted FGSM aims to push the dataset towards the feature class, and as seen in Figure 5.10a, this has a gradual negative impact on the ML metrics, as the value of epsilon increases towards the maximum value of 1.0. More specifically, the accuracy of the model drops to 42.1% from 95.6%, whilst its recall score drops to 49.3% from 95.7%. Using non-targeted FGSM generated adversarial examples, the increase of the epsilon value significantly impacts the ML metrics that decrease sharply after its value surpasses 0.1, as seen in Figure 5.10b. However, adversaries that exploit this attack may evade their specific attacks' detection but still be detected by the ML IDS,

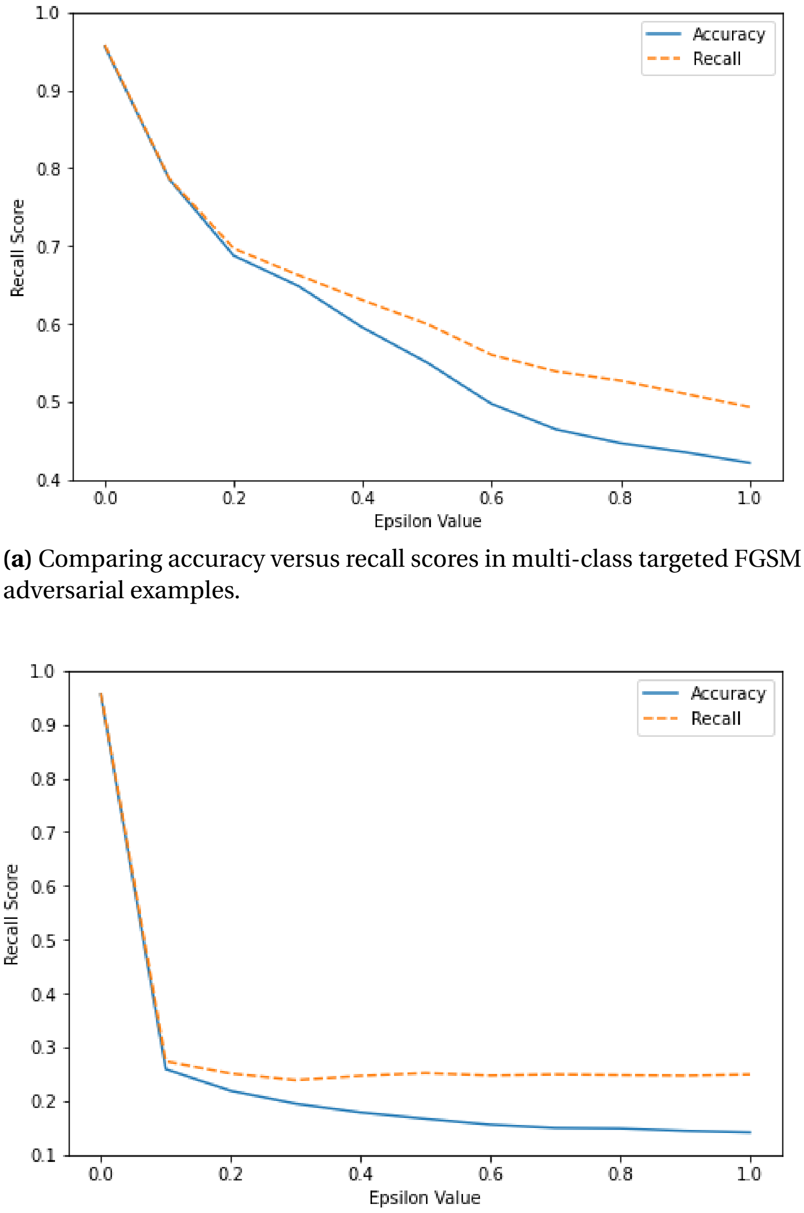


(a) Comparing accuracy versus recall scores in binary targeted FGSM adversarial examples.

(b) Comparing accuracy versus recall scores in binary non-targeted FGSM adversarial examples.

Figure 5.9: Comparing accuracy versus recall scores in binary targeted and non-targeted FGSM adversarial examples [69].

classifying their intentions as another type of attack and blocking them. By comparing the confusion matrices before and after manipulation, as seen in Figure 5.3b and Figure 5.7b, respectively, it can be observed that a considerable number of DoS and DDoS attacks is being classified as benign, whilst the false-negative rates of the other attacks such as reconnaissance and information theft remain low. The key finding from these experiments is that adversaries that launch DoS or DDoS attacks have higher chances of evading the ML IDS detection.



(a) Comparing accuracy versus recall scores in multi-class targeted FGSM adversarial examples.

(b) Comparing accuracy vs recall scores in multi-class non-targeted FGSM adversarial examples.

Figure 5.10: Comparing accuracy versus recall scores in five-class targeted and non-targeted FGSM adversarial examples [69].

5.3 Conclusion

ML IDS are increasing in popularity and are being used widely. However, their biggest threat is detection avoidance by adversaries. Additionally, the research interest related to adversarial examples also increased in the last few years. Adversarial examples aim to *trick* the ML models they exploit; hence, in the IDS domain, to trick the ML IDS into classifying adversary attempts as benign traffic [65, 343].

In this chapter, the conducted experiments demonstrated the impact of adversarial examples in a ML IDS scenario, utilising the Bot-IoT dataset [169]. As mentioned in the introduction of this chapter, this work is the first of its kind, investigated the Bot-IoT dataset [169] in adversarial settings. Firstly, an SVM model was developed to measure the impact of targeted and non-targeted label flipping. Secondly, two ANN submodels were developed, one for binary classification and one for multi-class, in this case, five-class classification. Key findings from these experiments revealed that various ML metrics mostly drop when affected by adversarial examples. More specifically, in the SVM model settings, the flipping of labels with the highest distance from the hyperplane affected the model the most instead of the initial indication that the lowest distance from the hyperplane would have the most significant impact. In the ANN settings, the targeted FGSM generated adversarial examples maintained a very high accuracy in the model, whilst its recall score was dropped significantly. Hence, organisations that employ such ML IDS models in their operations should also take into consideration the recall score in pair with the accuracy of the model.

Intriguing future avenues for the experiments presented in this chapter would be the investigation of adversarial examples in other IDS datasets, in which the attacks surface would be more balanced and not so heavily weighted towards DoS and DDoS traffic; however, this is currently a real-world limitation since the generation of mass balanced attacking traffic is eminently challenging. Further, another interesting future approach regarding the investigation of adversarial examples would be in ML models incorporated with other privacy-preserving ML techniques and countermeasures against them, such as adversarial training [182, 344].

This chapter investigates the impact of adversarial attacks from both an adversary's and the defender's perspectives. However, it should be noted that this chapter examined adversarial example attacks and not other crucial ML attacks such as the model inversion, which aims to reconstruct data used for the training of the models (Appendix E) [345]. The findings and experimentation presented in this chapter can be applied to other ML models and datasets in order to examine the impact of adversarial examples on other domains.

Sixth Chapter

Conclusions

Typically, technological advancements and systems, even in critical areas, focus on developing new features and capabilities, with their security and privacy often an afterthought. However, a shift towards the development of systems with security in mind since their conceptualisation has been observed. Some of these systems involve DLTs and blockchains, decentralisation of systems, and the invention of the privacy-preserving ML domain. The thesis discussed areas that can benefit from the presented innovative technologies, such as the IoT, Big Data and Cloud computing, alongside future directions to create a more secure world [69].

Blockchain technology has attracted much interest in the last few years. Primarily, this interest was focused on its financial aspects, the cryptocurrencies. However, it has evolved to benefit a broader spectrum of fields that can help solve previously non-investigated problems. This growth created similar DLT architectures focused on the fundamental underlying technology that is secure and private [13]. Furthermore, the emerging SSI concept is prevalent and is considered the future of the citizens' digital identities. The advantages for the data subjects, the citizens, cannot be neglected by organisations and governmental bodies since SSI transfers the true ownership of the generated data back to their owners. Since this concept can be easily combined with other ML and AI approaches, without adding significant computational overhead, the data subjects control which data and for how long they need to share with ML and AI models and revoke their access to it. However, these ML and AI approaches come with their own challenges, security and privacy concerns and careful consideration should be given when these approaches are being adopted in highly-sensitive environments.

The main findings and contributions of this thesis can be seen as follows:

- To address the **Objective I**, this thesis presented use cases of one of the most paramount private-permissioned DLTs, the Hyperledger Fabric, in the domain of the DNS. PRESERVE DNS has been presented, which enables the secure storage of passive DNS records and allows access to some of their private details, such as the IP addresses of the end-users that performed the DNS queries, only to authorised participants, the end-users themselves [66]. The security and performance of this system have been experimentally evaluated in comparison with other works in the literature, and it was demonstrated how it could be used on top of other related technologies.

- To address the **Objective II**, the previous contribution extended and applied to another critical domain such as healthcare, and PREHEALTH presented, a privacy-preserving EHR management system using the Hyperledger Fabric [67]. This system is tailored to a medical architecture with varying access levels to the stored data utilising the similar private data collection feature as shown in PRESERVE DNS [66]. As previously, PREHEALTH was experimentally evaluated in terms of security and performance against other works in the literature. The key findings of these experiments focused on the privacy-preservation and efficiency of systems that employ this technology without adding substantial computational overhead and improving current practices [66, 29, 67]. Additionally, these significant contributions and novelties to the current practices can be employed by existing infrastructures without requiring complete remodelling.

- To address the **Objective III**, this thesis presented a promising SSI solution that utilises digital identities DLTs that can be combined with other privacy-preserving ML and AI approaches [68]. This system presented a healthcare scenario with six participants, a ML researcher who aims to train a ML model from private data stored in each of the three participating hospitals' premises, an NHS Trust, and a governmental regulatory authority. In this scenario, the ML researcher is firstly required to acquire a digital participation credential from the governmental regulatory authority, as well as each of the three hospitals is required to acquire a similar digital participation credential from the NHS Trust. Further, to establish a secure connection, the researcher and each of the three hospitals must present their digital participation credential to the other party, who can verify it automatically using a digital identities blockchain ledger. For the verification of the credential, there is no need to contact the governmental regulatory authority or the NHS Trust since their private details are used to generate the participation credentials and the information stored in the digital identities blockchain ledger can be used to verify it automatically. After establishing the secure connection, the ML researcher can initiate a FL training by sending the ML model sequentially to each of the three hospitals to train it using their sensitive data. Hence, no raw, sensitive data transmission occurs at any point of the ML training. It should be noted that the improvement of the ML performance itself was left out of the scope of this experiment since it was focused on the demonstration that ML training is possible in a decentralised scenario as the aforementioned and can occur through the DIDComm channel. The evaluation of this system shows that its security has been dramatically improved since the ML models are being transmitted end-to-end encrypted, through the DIDComm channel between the participants, without compromising the system's performance in comparison with traditional FL approaches. It should be noted that, not surprisingly, a common challenge for these systems is the human factor itself. Since all the interactions occur based on the utilisation and presentation of various forms of digital credentials, their protection against adversaries is critical. A potential security breach and abuse of these credentials may result in major disruptions to the business continuity, as well as exfiltration and exploitation of sensitive data depending on the architecture of the affected system [68].

- Finally, to address the **Objective IV**, this thesis investigated, from an adversary's perspective, the impact of their attacks on ML IDS systems. The adversary's goal was to *trick* the ML IDS into classifying a malicious attempt as benign. This work focused on investigating these activities utilising a recent Bot-IoT dataset [169]. Hence, the key findings and contributions from these activities, on the one hand, highlighted the problem that is often neglected for enterprises that employ such systems and, on the other hand, revealed their impact that should be carefully considered, as well as potential defensive countermeasures to diminish or mitigate these issues [69]. The literature related to this topic has been very active in the last few years; however, a complete defensive countermeasure to mitigate this problem has not yet been found. Additionally, as presented, the BoT-IoT dataset is fairly imbalanced, focused more on some specific types of attacks; hence, the impact of adversarial attacks on a balanced dataset may have different outcomes and should be investigated more in the future.

6.1 Future Work

As future work for the presented approaches, technologies and systems individually can be found within the conclusion of each respective experimental chapter. However, as a whole, a combined system is envisioned that incorporates DLT, self-sovereign digital identities and ML/AI. The chosen DLT can store the sensitive data in a privacy-preserving manner, which only authorised participants can access. The access control policy and the authorisation of the participants can occur using self-sovereign digital identities stored in a public blockchain. The citizens who choose to use this system can control which data they want to share with the other infrastructure participants and can deny or revoke access to it easily and quickly. Finally, the security analysis of the stored data can occur using privacy-preserving ML and AI techniques, adding an extra layer of security and privacy. Regarding these ML and AI techniques, careful consideration should be given to a range of attacks against them, including but not limited to only adversarial examples, and countermeasures against them with few examples including adversarial training, differential privacy, and knowledge distillation.

However, the question that derives is whether we can build a secure world on top of an insecure one[1]. As the whole world becomes data-centric, the privacy of the end-users is remarkably valuable, thus motivating all future solutions to aim to preserve it.

[1] Bruce Schneier in Privacy, Trust and the Future at Edinburgh Napier University: `https://www.youtube.com/watch?v=eFmsCSIEMlw`

References

[1] Bruce Schneier. *Privacy, Trust and the Future. Edinburgh Napier University*. Accessed on 01 Mar 2022. 2019. URL: `https://www.youtube.com/watch?v=eFmsCSIEMlw`.

[2] Suchet Sapre, Pouyan Ahmadi and Khondkar Islam. 'A Robust Comparison of the KDDCup99 and NSL-KDD IoT Network Intrusion Detection Datasets Through Various Machine Learning Algorithms'. In: *arXiv preprint arXiv:1912.13204* (2019).

[3] Chih-Liang Yeh. 'Pursuing consumer empowerment in the age of big data: A comprehensive regulatory framework for data brokers'. In: *Telecommunications Policy* 42.4 (2018), pp. 282–292.

[4] Paul Voigt and Axel Von dem Bussche. 'The EU General Data Protection Regulation (GDPR)'. In: *A Practical Guide, 1st Ed., Cham: Springer International Publishing* (2017).

[5] Spyros Kokolakis. 'Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon'. In: *Computers & security* 64 (2017), pp. 122–134.

[6] Satoshi Nakamoto et al. 'Bitcoin: A peer-to-peer electronic cash system'. In: (2008).

[7] Massimo Di Pierro. 'What is the blockchain?' In: *Computing in Science & Engineering* 19.5 (2017), pp. 92–95.

[8] Lou Carlozo. 'What is blockchain?' In: *Journal of Accountancy* 224.1 (2017), p. 29.

[9] Wenbo Wang, Dinh Thai Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang Wen and Dong In Kim. 'A survey on consensus mechanisms and mining strategy management in blockchain networks'. In: *Ieee Access* 7 (2019), pp. 22328–22370.

[10] Fran Casino, Thomas K Dasaklis and Constantinos Patsakis. 'A systematic literature review of blockchain-based applications: current status, classification and open issues'. In: *Telematics and Informatics* 36 (2019), pp. 55–81.

[11] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen and Huaimin Wang. 'Blockchain challenges and opportunities: A survey'. In: *International Journal of Web and Grid Services* 14.4 (2018), pp. 352–375.

[12] Gavin Wood et al. 'Ethereum: A secure decentralised generalised transaction ledger'. In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

[13] Elli Androulaki et al. 'Hyperledger fabric: a distributed operating system for permissioned blockchains'. In: *Proceedings of the Thirteenth EuroSys Conference*. ACM. 2018, p. 30.

[14] Akanksha Kaushik, Archana Choudhary, Chinmay Ektare, Deepti Thomas and Syed Akram. 'Blockchain—Literature survey'. In: *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE. 2017, pp. 2145–2148.

[15] Ralph C Merkle. 'A digital signature based on a conventional encryption function'. In: *Conference on the theory and application of cryptographic techniques*. Springer. 1987, pp. 369–378.

[16] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen and Huaimin Wang. 'An overview of blockchain technology: Architecture, consensus, and future trends'. In: *2017 IEEE international congress on big data (BigData congress)*. IEEE. 2017, pp. 557–564.

[17] Christian Cachin et al. ‘Architecture of the hyperledger blockchain fabric’. In: *Workshop on distributed cryptocurrencies and consensus ledgers*. Vol. 310. 2016, p. 4.

[18] R Elmasri, Shamkant B Navathe, R Elmasri and SB Navathe. *Fundamentals of Database Systems*. Springer, 2000.

[19] Tsung-Ting Kuo, Hyeon-Eui Kim and Lucila Ohno-Machado. ‘Blockchain distributed ledger technologies for biomedical and health care applications’. In: *Journal of the American Medical Informatics Association* 24.6 (2017), pp. 1211–1220.

[20] Lakshmi Siva Sankar, M Sindhu and M Sethumadhavan. ‘Survey of consensus protocols on blockchain applications’. In: *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE. 2017, pp. 1–5.

[21] P Kravchenko. ‘Ok, I need a blockchain, but which one’. In: *Medium. Available online at: https://medium. com/@ pavelkravchenko/ca75c1e2100* (2016). Accessed on 01 Mar 2022.

[22] Yang Xiao, Ning Zhang, Wenjing Lou and Y Thomas Hou. ‘A survey of distributed consensus protocols for blockchain networks’. In: *IEEE Communications Surveys & Tutorials* 22.2 (2020), pp. 1432–1465.

[23] Arati Baliga. ‘Understanding blockchain consensus models’. In: *Persistent*. 2017.

[24] Natalia Chaudhry and Muhammad Murtaza Yousaf. ‘Consensus algorithms in blockchain: comparative analysis, challenges and opportunities’. In: *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*. IEEE. 2018, pp. 54–63.

[25] Giang-Truong Nguyen and Kyungbaek Kim. ‘A Survey about Consensus Algorithms Used in Blockchain.’ In: *Journal of Information processing systems* 14.1 (2018).

[26] Stefano De Angelis, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri and Vladimiro Sassone. 'PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain'. In: *Italian Conference on Cyber Security (06/02/18)*. Accessed on 01 Mar 2022. 2018. URL: https://eprints.soton.ac.uk/415083/.

[27] LM Bach, Branko Mihaljevic and Mario Zagar. 'Comparative analysis of blockchain consensus algorithms'. In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE. 2018, pp. 1545–1550.

[28] Rui Zhang, Rui Xue and Ling Liu. 'Security and privacy on blockchain'. In: *ACM Computing Surveys (CSUR)* 52.3 (2019), pp. 1–34.

[29] Pavlos Papadopoulos, Nikolaos Pitropakis and William J. Buchanan. 'Decentralized Privacy: A Distributed Ledger Approach'. In: *Handbook of Smart Materials, Technologies, and Devices: Applications of Industry 4.0*. Ed. by Chaudhery Mustansar Hussain and Paolo Di Sia. Cham: Springer International Publishing, 2020, pp. 1–26. ISBN: 978-3-030-58675-1. DOI: 10.1007/978-3-030-58675-1_58-2. URL: https://doi.org/10.1007/978-3-030-58675-1_58-2.

[30] Stefan Poslad. *Ubiquitous computing: smart devices, environments and interactions*. John Wiley & Sons, 2011.

[31] Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag and Mikael Gidlund. 'Industrial internet of things: Challenges, opportunities, and directions'. In: *IEEE Transactions on Industrial Informatics* 14.11 (2018), pp. 4724–4734.

[32] Laurence Goasduff. *Gartner Predicts Outdoor Surveillance Cameras Will Be Largest Market for 5G Internet of Things Solutions Over Next Three Years*. Accessed on 01 Mar 2022. 2019. URL: https://www.gartner.com/en/newsroom/press-releases/2019-10-17-gartner-predicts-outdoor-surveillance-cameras-will-be.

[33] Olakunle Ibitoye, Omair Shafiq and Ashraf Matrawy. ‘Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks’. In: *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2019, pp. 1–6.

[34] Bjoern M Eskofier et al. ‘Recent machine learning advancements in sensor-based mobility analysis: Deep learning for Parkinson’s disease assessment’. In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2016, pp. 655–658.

[35] Mohammad Al-Rubaie and J Morris Chang. ‘Privacy-preserving machine learning: Threats and solutions’. In: *IEEE Security & Privacy* 17.2 (2019), pp. 49–58.

[36] Alexander Mühle, Andreas Grüner, Tatiana Gayvoronskaya and Christoph Meinel. ‘A survey on essential components of a self-sovereign identity’. In: *Computer Science Review* 30 (2018), pp. 80–86.

[37] Pádraig Cunningham, Matthieu Cord and Sarah Jane Delany. ‘Supervised learning’. In: *Machine learning techniques for multimedia*. Springer, 2008, pp. 21–49.

[38] Horace B Barlow. ‘Unsupervised learning’. In: *Neural computation* 1.3 (1989), pp. 295–311.

[39] Xiaojin Jerry Zhu. ‘Semi-supervised learning literature survey’. In: (2005).

[40] Ilemona S Atawodi. ‘A Machine Learning Approach to Network Intrusion Detection System Using K Nearest Neighbor and Random Forest’. In: *Master’s Theses* 651 (2019).

[41] Antonia Nisioti, Alexios Mylonas, Paul D Yoo and Vasilios Katos. ‘From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods’. In: *IEEE Communications Surveys & Tutorials* 20.4 (2018), pp. 3369–3388.

[42] Fabian Pedregosa et al. ‘Scikit-learn: Machine learning in Python’. In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[43] Orestis Christou, Nikolaos Pitropakis, Pavlos Papadopoulos, Sean McKeown and William J Buchanan. 'Phishing URL Detection Through Top-level Domain Analysis: A Descriptive Approach'. In: *arXiv preprint arXiv:2005.06599* (2020).

[44] Debi Prasanna Acharjya and K Ahmed. 'A survey on big data analytics: challenges, open research issues and tools'. In: *International Journal of Advanced Computer Science and Applications* 7.2 (2016), pp. 511–518.

[45] Balaji Balakrishnan. *Cloud Security Monitoring*. Accessed on 01 Mar 2022. 2017. URL: https://www.sans.org/reading-room/whitepapers/cloud/paper/37672.

[46] Peter Kairouz et al. 'Advances and open problems in federated learning'. In: *arXiv preprint arXiv:1912.04977* (2019).

[47] Will Abramson, Adam James Hall, Pavlos Papadopoulos, Nikolaos Pitropakis and William J Buchanan. 'A Distributed Trust Framework for Privacy-Preserving Machine Learning'. In: *International Conference on Trust and Privacy in Digital Business*. Springer. 2020, pp. 205–220.

[48] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal and Karn Seth. 'Practical Secure Aggregation for Federated Learning on User-Held Data'. In: *CoRR* abs/1611.04482 (2016). arXiv: 1611.04482. URL: http://arxiv.org/abs/1611.04482.

[49] Ivan Kholod, Evgeny Yanaki, Dmitry Fomichev, Evgeniy Shalugin, Evgenia Novikova, Evgeny Filippov and Mats Nordlund. 'Open-Source Federated Learning Frameworks for IoT: A Comparative Review and Analysis'. In: *Sensors* 21.1 (2021), p. 167.

[50] Otkrist Gupta and Ramesh Raskar. 'Distributed learning of deep neural network over multiple agents'. In: *Journal of Network and Computer Applications* 116 (2018), pp. 1–8.

[51] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish and Ramesh Raskar. 'Split learning for health: Distributed deep learning without sharing raw patient data'. In: *arXiv preprint arXiv:1812.00564* (2018).

[52] Praneeth Vepakomma, Tristan Swedish, Ramesh Raskar, Otkrist Gupta and Abhimanyu Dubey. *No Peek: A Survey of private distributed deep learning.* 2018. arXiv: `1812.03288 [cs.LG]`.

[53] Praneeth Vepakomma, Chetan Tonde, Ahmed Elgammal et al. 'Supervised dimensionality reduction via distance correlation maximization'. In: *Electronic Journal of Statistics* 12.1 (2018), pp. 960–984.

[54] Praneeth Vepakomma, Otkrist Gupta, Abhimanyu Dubey and Ramesh Raskar. 'Reducing leakage in distributed deep learning for sensitive health data'. In: *arXiv preprint arXiv:1812.00564* (2019).

[55] Gábor J. Székely, Maria L. Rizzo and Nail K. Bakirov. 'Measuring and testing dependence by correlation of distances'. In: *Ann. Statist.* 35.6 (Dec. 2007), pp. 2769–2794. DOI: `10.1214/009053607000000505`. URL: `https://doi.org/10.1214/009053607000000505`.

[56] Dan Bogdanov, Riivo Talviste and Jan Willemson. 'Deploying secure multi-party computation for financial data analysis'. In: *International Conference on Financial Cryptography and Data Security.* Springer. 2012, pp. 57–64.

[57] Peter Bogetoft et al. 'Secure multiparty computation goes live'. In: *International Conference on Financial Cryptography and Data Security.* Springer. 2009, pp. 325–343.

[58] Cynthia Dwork. 'Differential privacy: A survey of results'. In: *International conference on theory and applications of models of computation.* Springer. 2008, pp. 1–19.

[59] Cynthia Dwork, Aaron Roth et al. 'The algorithmic foundations of differential privacy'. In: *Foundations and Trends in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.

[60] Cynthia Dwork, Frank McSherry, Kobbi Nissim and Adam Smith. 'Calibrating Noise to Sensitivity in Private Data Analysis'. In: *Theory of Cryptography*. Ed. by Shai Halevi and Tal Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284. ISBN: 978-3-540-32732-5.

[61] Apple. *Private Federated Learning (NeurIPS 2019 Expo Talk Abstract)*. Accessed on 01 Mar 2022. 2019. URL: `https://nips.cc/ExpoConferences/2019/schedule?talk_id=40`.

[62] Craig Gentry. 'Fully homomorphic encryption using ideal lattices'. In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 2009, pp. 169–178.

[63] Ryszard Stanislaw Michalski, Jaime Guillermo Carbonell and Tom M Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[64] Yang Liu, Debiao He, Mohammad S Obaidat, Neeraj Kumar, Muhammad Khurram Khan and Kim-Kwang Raymond Choo. 'Blockchain-based identity management systems: A review'. In: *Journal of network and computer applications* 166 (2020), p. 102731.

[65] Alexey Kurakin, Ian Goodfellow and Samy Bengio. 'Adversarial machine learning at scale'. In: *arXiv preprint arXiv:1611.01236* (2016).

[66] Pavlos Papadopoulos, Nikolaos Pitropakis, William J Buchanan, Owen Lo and Sokratis Katsikas. 'Privacy-Preserving Passive DNS'. In: *Computers* 9.3 (2020), p. 64.

[67] Charalampos Stamatellis, Pavlos Papadopoulos, Nikolaos Pitropakis, Sokratis Katsikas and William J Buchanan. 'A Privacy-Preserving Healthcare Framework Using Hyperledger Fabric'. In: *Sensors* 20.22 (2020), p. 6587.

[68] Pavlos Papadopoulos, Will Abramson, Adam J Hall, Nikolaos Pitropakis and William J Buchanan. 'Privacy and trust redefined in federated machine learning'. In: *Machine Learning and Knowledge Extraction* 3.2 (2021), pp. 333–356.

[69] Pavlos Papadopoulos, Oliver Thornewill von Essen, Nikolaos Pitropakis, Christos Chrysoulas, Alexios Mylonas and William J. Buchanan. ‘Launching Adversarial Attacks against Network Intrusion Detection Systems for IoT’. In: *Journal of Cybersecurity and Privacy* 1.2 (2021), pp. 252–273. ISSN: 2624-800X. DOI: 10.3390/jcp1020014. URL: http://dx.doi.org/10.3390/jcp1020014.

[70] Barbara A Kitchenham, Shari Lawrence Pfleeger, Lesley M Pickard, Peter W Jones, David C. Hoaglin, Khaled El Emam and Jarrett Rosenberg. ‘Preliminary guidelines for empirical research in software engineering’. In: *IEEE Transactions on software engineering* 28.8 (2002), pp. 721–734.

[71] Jakub Konecny, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh and Dave Bacon. ‘Federated learning: Strategies for improving communication efficiency’. In: *arXiv preprint arXiv:1610.05492* (2016).

[72] Barbara Kitchenham and Stuart Charters. ‘Guidelines for performing systematic literature reviews in software engineering’. In: (2007).

[73] Hossein Shirazi, Bruhadeshwar Bezawada and Indrakshi Ray. ‘" Kn0w Thy Doma1n Name" Unbiased Phishing Detection Using Domain Name Based Features’. In: *Proceedings of the 23nd ACM on symposium on access control models and technologies*. 2018, pp. 69–75.

[74] Kaan Onarlioglu, Utku Ozan Yilmaz, Engin Kirda and Davide Balzarotti. ‘Insights into User Behavior in Dealing with Internet Attacks.’ In: *NDSS*. 2012.

[75] Nick Nikiforakis, Marco Balduzzi, Lieven Desmet, Frank Piessens and Wouter Joosen. ‘Soundsquatting: Uncovering the use of homophones in domain squatting’. In: *International Conference on Information Security*. Springer. 2014, pp. 291–308.

[76] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gomez, Nikolaos Pitropakis, Nick Nikiforakis and Manos Antonakakis. ‘Hiding in plain sight: A longitudinal study of combosquatting abuse’. In: *Pro-*

ceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM. 2017, pp. 569–586.

[77] Ankit Kumar Jain and Brij B Gupta. 'Phishing detection: analysis of visual similarity based approaches'. In: *Security and Communication Networks* 2017 (2017).

[78] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi and Chris Kanich. 'The Long {"Taile"} of Typosquatting Domain Names'. In: *23rd USENIX Security Symposium (USENIX Security 14)*. 2014, pp. 191–206.

[79] Proofpoint. *THE HUMAN FACTOR: PEOPLE CENTERED THREATS DEFINE THE LANDSCAPE*. Accessed on 01 Mar 2022. 2018. URL: `https://www.key4biz.it/wp-content/uploads/2018/04/pfpt-us-wp-human-factor-report-2018-180425.pdf`.

[80] Abdallah Moubayed, MohammadNoor Injadat, Abdallah Shami and Hanan Lutfiyya. 'DNS Typo-Squatting Domain Detection: A Data Analytics & Machine Learning Based Approach'. In: *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2018, pp. 1–7.

[81] Gunter Ollmann. 'The Phishing Guide–Understanding & Preventing Phishing Attacks'. In: *NGS Software Insight Security Research* (2004).

[82] Ankit Kumar Jain and BB Gupta. 'A survey of phishing attack techniques, defence mechanisms and open research challenges'. In: *Enterprise Information Systems* 16.4 (2022), pp. 527–565.

[83] Verizon. *2019 Data Breach Investigations Report*. 2019. URL: `http://www.sciencedirect.com/science/article/pii/S1361372319300600`.

[84] Verizon. *2018 Data Breach Investigations Report*. Accessed on 01 Mar 2022. 2018. URL: `https://enterprise.verizon.com/resources/reports/dbir/rp_data-breach-investigations-report-2013_en_xg.pdf`.

[85] Maria Vergelis and Tatyana Shcherbakova. *Spam and phishing in Q1 2019*. Accessed on 01 Mar 2022. 2019. URL: `https://securelist.com/spam-and-phishing-in-q1-%202019/90795/`.

[86] Kang Leng Chiew, Kelvin Sheng Chek Yong and Choon Lin Tan. 'A survey of phishing attacks: Their types, vectors and technical approaches'. In: *Expert Systems with Applications* 106 (2018), pp. 1–20.

[87] Tian Lin, Daniel E Capecci, Donovan M Ellis, Harold A Rocha, Sandeep Dommaraju, Daniela S Oliveira and Natalie C Ebner. 'Susceptibility to spear-phishing emails: Effects of internet user demographics and email content'. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 26.5 (2019), pp. 1–28.

[88] Richard G Brody, Elizabeth Mulig and Valerie Kimball. 'PHISHING, PHARMING AND IDENTITY THEFT.' In: *Academy of Accounting & Financial Studies Journal* 11.3 (2007).

[89] Florian Weimer. 'Passive DNS replication'. In: *FIRST conference on computer security incident*. 2005, p. 98.

[90] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee and Nick Feamster. 'Building a dynamic reputation system for dns.' In: *USENIX security symposium*. 2010, pp. 273–290.

[91] Leyla Bilge, Engin Kirda, Christopher Kruegel and Marco Balduzzi. 'EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis.' In: *Ndss*. 2011, pp. 1–17.

[92] Issa Khalil, Ting Yu and Bei Guan. 'Discovering malicious domains through passive DNS data graph analysis'. In: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM. 2016, pp. 663–674.

[93] Chaz Lever, Robert Walls, Yacin Nadji, David Dagon, Patrick McDaniel and Manos Antonakakis. 'Domain-Z: 28 registrations later measuring the exploitation of residual trust in domains'. In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, pp. 691–706.

[94] Omar Alrawi, Chaz Lever, Manos Antonakakis and Fabian Monrose. 'Sok: Security evaluation of home-based iot deployments'. In: *IEEE S&P*. 2019, pp. 208–226.

[95] Farsight Security. *DNSDB*. Accessed on 01 Mar 2022. 2010. URL: `https://www.farsightsecurity.com/solutions/dnsdb/`.

[96] VirusTotal. *VirusTotal Passive DNS replication*. Accessed on 01 Mar 2022. 2013. URL: `https://www.virustotal.com/gui/home/search`.

[97] Loren Weith, Deepen Desai and Amit Sinha. *Cloud based systems and methods for determining security risks of users and groups*. US Patent App. 10/142,362. 2018.

[98] Ke Tian, Steve TK Jan, Hang Hu, Danfeng Yao and Gang Wang. 'Needle in a haystack: tracking down elite phishing domains in the wild'. In: *Proceedings of the Internet Measurement Conference 2018*. ACM. 2018, pp. 429–442.

[99] Egon Kidmose, Erwin Lansing, Soren Brandbyge and Jens Myrup Pedersen. 'Detection of malicious and abusive domain names'. In: *2018 1st International Conference on Data Intelligence and Security (ICDIS)*. IEEE. 2018, pp. 49–56.

[100] Paolo Piredda, Davide Ariu, Battista Biggio, Igino Corona, Luca Piras, Giorgio Giacinto and Fabio Roli. 'Deepsquatting: Learning-based typosquatting detection at deeper domain levels'. In: *Conference of the Italian Association for Artificial Intelligence*. Springer. 2017, pp. 347–358.

[101] Jose Selvi, Ricardo J Rodriguez and Emilio Soria-Olivas. 'Detection of algorithmically generated malicious domain names using masked N-grams'. In: *Expert Systems with Applications* 124 (2019), pp. 156–163.

[102] Haya Shulman. 'Pretty Bad Privacy: Pitfalls of DNS Encryption.' In: *WPES*. 2014, pp. 191–200.

[103] Supranamaya Ranjan. *Detecting DNS fast-flux anomalies*. US Patent 8,260,914. 2012.

[104] Georgios Kambourakis, Tassos Moschos, Dimitris Geneiatakis and Stefanos Gritzalis. 'Detecting DNS amplification attacks'. In: *International Workshop on Critical Information Infrastructures Security*. Springer. 2007, pp. 185–196.

[105] Sooel Son and Vitaly Shmatikov. 'The hitchhiker's guide to DNS cache poisoning'. In: *International Conference on Security and Privacy in Communication Systems.* Springer. 2010, pp. 466–483.

[106] Joe Stewart. *DNS cache poisoning–the next generation.* 2003.

[107] Jonathan M Spring and Carly L Huth. 'The impact of passive dns collection on end-user privacy'. In: *Securing and Trusting Internet Names* (2012).

[108] Bojan Zdrnja. 'Security Monitoring of DNS traffic'. In: *University of Auckland* (2006).

[109] Jivesh Govil and Jivika Govil. '4G mobile communication systems: Turns, trends and transition'. In: *2007 International Conference on Convergence Information Technology (ICCIT 2007).* IEEE. 2007, pp. 13–18.

[110] Jun Xu, Jinliang Fan, Mostafa H Ammar and Sue B Moon. 'Prefix-preserving ip address anonymization: Measurement-based security evaluation and a new cryptography-based scheme'. In: *10th IEEE International Conference on Network Protocols, 2002. Proceedings.* IEEE. 2002, pp. 280–289.

[111] Athanasios Kountouras, Panagiotis Kintis, Chaz Lever, Yizheng Chen, Yacin Nadji, David Dagon, Manos Antonakakis and Rodney Joffe. 'Enabling network security through active DNS datasets'. In: *International Symposium on Research in Attacks, Intrusions, and Defenses.* Springer. 2016, pp. 188–208.

[112] Jingqiang Liu, Bin Li, Lizhang Chen, Meng Hou, Feiran Xiang and Peijun Wang. 'A Data Storage Method Based on Blockchain for Decentralization DNS'. In: *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC).* IEEE. 2018, pp. 189–196.

[113] Xueping Liang, Sachin Shetty, Deepak Tosh, Charles Kamhoua, Kevin Kwiat and Laurent Njilla. 'Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability'. In: *Proceedings of the 17th IEEE/ACM international symposium on cluster, cloud and grid computing.* IEEE Press. 2017, pp. 468–477.

[114] Harry A Kalodner, Miles Carlsten, Paul Ellenbogen, Joseph Bonneau and Arvind Narayanan. 'An Empirical Study of Namecoin and Lessons for Decentralized Namespace Design.' In: *WEIS*. Citeseer. 2015.

[115] Muneeb Ali, Jude Nelson, Ryan Shea and Michael J Freedman. 'Blockstack: A global naming and storage system secured by blockchains'. In: *2016 USENIX Annual Technical Conference (USENIX ATC 16)*. 2016, pp. 181–194.

[116] Zhong Yu, Dong Xue, Jiulun Fan and Chang Guo. 'DNSTSM: DNS Cache Resources Trusted Sharing Model Based on Consortium Blockchain'. In: *IEEE Access* 8 (2020), pp. 13640–13650.

[117] Ajay Kumar and Sunita Garhwal. 'State-of-the-Art Survey of Quantum Cryptography'. In: *Archives of Computational Methods in Engineering* 28.5 (2021), pp. 3831–3868.

[118] Erin English, Amy Davine Kim and Michael Nonaka. *Advancing Blockchain Cybersecurity: Technical and Policy Considerations for the Financial Services Industry*. Accessed on 01 Mar 2022. 2018. URL: `https://www.microsoft.com/en-us/cybersecurity/content-hub/advancing-blockchain-cybersecurity`.

[119] Xabier Larrucea, Micha Moffie, Sigal Asaf and Izaskun Santamaria. 'Towards a GDPR compliant way to secure European cross border Healthcare Industry 4.0'. In: *Computer Standards & Interfaces* 69 (2020), p. 103408.

[120] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Schumacher and Fusheng Wang. 'Secure and trustable electronic medical records sharing using blockchain'. In: *AMIA annual symposium proceedings*. Vol. 2017. American Medical Informatics Association. 2017, p. 650.

[121] Marko Hölbl, Marko Kompara, Aida Kamišalić and Lili Nemec Zlatolas. 'A systematic review of the use of blockchain in healthcare'. In: *Symmetry* 10.10 (2018), p. 470.

[122] André Henrique Mayer, Cristiano André da Costa and Rodrigo da Rosa Righi. 'Electronic health records in a blockchain: a systematic review'. In: *Health informatics journal* 26.2 (2020), pp. 1273–1288.

[123] Buket Yüksel, Alptekin Küpçü and Öznur Özkasap. 'Research issues for privacy and security of electronic health services'. In: *Future Generation Computer Systems* 68 (2017), pp. 1–13.

[124] Thomas Bocek, Bruno B Rodrigues, Tim Strasser and Burkhard Stiller. 'Blockchains everywhere-a use-case of blockchains in the pharma supply-chain'. In: *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE. 2017, pp. 772–777.

[125] Assad Abbas and Samee U Khan. 'A review on the state-of-the-art privacy-preserving approaches in the e-health clouds'. In: *IEEE Journal of Biomedical and Health Informatics* 18.4 (2014), pp. 1431–1441.

[126] Elisa Bertino and Ravi Sandhu. 'Database security-concepts, approaches, and challenges'. In: *IEEE Transactions on Dependable and secure computing* 2.1 (2005), pp. 2–19.

[127] Franco Callegati, Walter Cerroni and Marco Ramilli. 'Man-in-the-Middle Attack to the HTTPS Protocol'. In: *IEEE Security & Privacy* 7.1 (2009), pp. 78–81.

[128] Alevtina Dubovitskaya, Visara Urovi, Matteo Vasirani, Karl Aberer and Michael I Schumacher. 'A cloud-based ehealth architecture for privacy preserving data integration'. In: *IFIP International Information Security and Privacy Conference*. Springer. 2015, pp. 585–598.

[129] Alex Roehrs, Cristiano André da Costa and Rodrigo da Rosa Righi. 'OmniPHR: A distributed architecture model to integrate personal health records'. In: *Journal of biomedical informatics* 71 (2017), pp. 70–81.

[130] Rui Guo, Huixian Shi, Qinglan Zhao and Dong Zheng. 'Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems'. In: *IEEE access* 6 (2018), pp. 11676–11686.

[131] Huawei Zhao, Yong Zhang, Yun Peng and Ruzhi Xu. 'Lightweight backup and efficient recovery scheme for health blockchain keys'. In: *2017 IEEE 13th International Symposium on autonomous decentralized system (ISADS)*. IEEE. 2017, pp. 229–234.

[132] Vishal Patel. 'A framework for secure and decentralized sharing of medical imaging data via blockchain consensus'. In: *Health informatics journal* 25.4 (2019), pp. 1398–1411.

[133] Daisuke Ichikawa, Makiko Kashiyama and Taro Ueno. 'Tamper-resistant mobile health using blockchain technology'. In: *JMIR mHealth and uHealth* 5.7 (2017), e111.

[134] Xueping Liang, Juan Zhao, Sachin Shetty, Jihong Liu and Danyi Li. 'Integrating blockchain for data sharing and collaboration in mobile healthcare applications'. In: *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*. IEEE. 2017, pp. 1–5.

[135] Asaph Azaria, Ariel Ekblaw, Thiago Vieira and Andrew Lippman. 'Medrec: Using blockchain for medical data access and permission management'. In: *2016 2nd International Conference on Open and Big Data (OBD)*. IEEE. 2016, pp. 25–30.

[136] Abdullah Al Omar, Mohammad Shahriar Rahman, Anirban Basu and Shinsaku Kiyomoto. 'Medibchain: A blockchain based privacy preserving platform for healthcare data'. In: *International conference on security, privacy and anonymity in computation, communication and storage*. Springer. 2017, pp. 534–543.

[137] Huihui Yang and Bian Yang. 'A blockchain-based approach to the secure sharing of healthcare data'. In: *Nisk Journal* (2017), pp. 100–111.

[138] Salvatore J Stolfo, Steven M Bellovin, Shlomo Hershkop, Angelos D Keromytis, Sara Sinclair and Sean W Smith. *Insider attack and cyber security: beyond the hacker*. Vol. 39. Springer Science & Business Media, 2008.

[139] QI Xia, Emmanuel Boateng Sifah, Kwame Omono Asamoah, Jianbin Gao, Xiaojiang Du and Mohsen Guizani. ‘MeDShare: Trust-less medical data sharing among cloud service providers via blockchain’. In: *IEEE Access* 5 (2017), pp. 14757–14767.

[140] Drew Ivan. ‘Moving toward a blockchain-based method for the secure storage of patient records’. In: *ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, Maryland, United States: ONC/NIST*. 2016, pp. 1–11.

[141] A Albeyatti. ‘White paper: medicalchain’. In: *MedicalChain self-publication* (2018).

[142] Alexander McLeod and Diane Dolezel. ‘Cyber-analytics: Modeling factors associated with healthcare data breaches’. In: *Decision Support Systems* 108 (2018), pp. 57–68.

[143] Abhishek Kumar Pandey, Asif Irshad Khan, Yoosef B Abushark, Md Mottahir Alam, Alka Agrawal, Rajeev Kumar and Raees Ahmad Khan. ‘Key Issues in Healthcare Data Integrity: Analysis and Recommendations’. In: *IEEE Access* 8 (2020), pp. 40612–40628.

[144] Lynne Coventry and Dawn Branley. ‘Cybersecurity in healthcare: a narrative review of trends, threats and ways forward’. In: *Maturitas* 113 (2018), pp. 48–52.

[145] William Smart. ‘Lessons learned review of the WannaCry ransomware cyber attack’. In: *Department of Health and Social Care, England UK, London* 1.20175 (2018), pp. 10–1038.

[146] M Alvarez. ‘Security trends in the healthcare industry’. In: *Somers: IBM* (2017), pp. 2–18.

[147] Jason RC Nurse, Oliver Buckley, Philip A Legg, Michael Goldsmith, Sadie Creese, Gordon RT Wright and Monica Whitty. ‘Understanding insider threat: A framework for characterising attacks’. In: *2014 IEEE security and privacy workshops*. IEEE. 2014, pp. 214–228.

[148] Ali Dorri, Salil S Kanhere, Raja Jurdak and Praveen Gauravaram. 'Blockchain for IoT security and privacy: The case study of a smart home'. In: *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*. IEEE. 2017, pp. 618–623.

[149] Dalmacio V Posadas Jr. 'The internet of things: the GDPR and the Blockchain may be incompatible'. In: *Journal of Internet Law* 21.11 (2018), pp. 1–29.

[150] Miranda Mourby, Elaine Mackey, Mark Elliot, Heather Gowans, Susan E Wallace, Jessica Bell, Hannah Smith, Stergios Aidinlis and Jane Kaye. 'Are 'pseudonymised'data always personal data? Implications of the GDPR for administrative data research in the UK'. In: *Computer Law & Security Review* 34.2 (2018), pp. 222–233.

[151] Christopher Kelly, Nikolaos Pitropakis, Sean McKeown and Costas Lambrinoudakis. 'Testing And Hardening IoT Devices Against the Mirai Botnet'. In: *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. IEEE. 2020, pp. 1–8.

[152] Daniel J Solove. *Nothing to hide: The false tradeoff between privacy and security*. Yale University Press, 2011.

[153] Federal Trade Commission et al. 'Protecting consumer privacy in an era of rapid change: Recommendations for businesses and policymakers'. In: *Washington, DC: Federal Trade Commission* (2012).

[154] Rodrigo Roman, Javier Lopez and Masahiro Mambo. 'Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges'. In: *Future Generation Computer Systems* 78 (2018), pp. 680–698.

[155] Nikolaos Pitropakis, Sokratis Katsikas and Costas Lambrinoudakis. 'Cloud Security, Privacy and Trust Baselines'. In: *Cloud Computing Security: Foundations and Challenges* (2020), p. 45.

[156] Lisa Muller, Christos Chrysoulas, Nikolaos Pitropakis and Peter J Barclay. ‘A Traffic Analysis on Serverless Computing Based on the Example of a File Upload Stream on AWS Lambda’. In: *Big Data and Cognitive Computing* 4.4 (2020), p. 38.

[157] Christopher Kelly, Nikolaos Pitropakis, Alexios Mylonas, Sean McKeown and William J Buchanan. ‘A Comparative Analysis of Honeypots on Different Cloud Platforms’. In: *Sensors* 21.7 (2021), p. 2433.

[158] David Bernstein. ‘Containers and cloud: From lxc to docker to kubernetes’. In: *IEEE Cloud Computing* 1.3 (2014), pp. 81–84.

[159] Nguyen Thanh Van, Tran Ngoc Thinh et al. ‘An anomaly-based network intrusion detection system using deep learning’. In: *2017 International Conference on System Science and Engineering (ICSSE)*. IEEE. 2017, pp. 210–214.

[160] Yan Naung Soe, Yaokai Feng, Paulus Insap Santosa, Rudy Hartanto and Kouichi Sakurai. ‘Towards a Lightweight Detection System for Cyber Attacks in the IoT Environment Using Corresponding Features’. In: *Electronics* 9.1 (2020), p. 144.

[161] Jesse Davis and Mark Goadrich. ‘The relationship between Precision-Recall and ROC curves’. In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 233–240.

[162] Peter A Flach. ‘The geometry of ROC space: understanding machine learning metrics through ROC isometrics’. In: *Proceedings of the 20th international conference on machine learning (ICML-03)*. 2003, pp. 194–201.

[163] Mohamed Faisal Elrawy, Ali Ismail Awad and Hesham FA Hamed. ‘Intrusion detection systems for IoT-based smart environments: a survey’. In: *Journal of Cloud Computing* 7.1 (2018), pp. 1–20.

[164] KDD Cup. ‘Data (1999)’. In: *URL http://www. kdd. org/kdd-cup/view/kdd-cup-1999/Data* (1999).

[165] John McHugh. ‘Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln

laboratory'. In: *ACM Transactions on Information and System Security (TISSEC)* 3.4 (2000), pp. 262–294.

[166] Matthew V Mahoney and Philip K Chan. 'An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection'. In: *International Workshop on Recent Advances in Intrusion Detection*. Springer. 2003, pp. 220–237.

[167] Nicholas Athanasiades, Randal Abler, John Levine, Henry Owen and George Riley. 'Intrusion detection testing and benchmarking methodologies'. In: *First IEEE International Workshop on Information Assurance, 2003. IWIAS 2003. Proceedings.* IEEE. 2003, pp. 63–72.

[168] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu and Ali A Ghorbani. 'A detailed analysis of the KDD CUP 99 data set'. In: *2009 IEEE symposium on computational intelligence for security and defense applications*. IEEE. 2009, pp. 1–6.

[169] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova and Benjamin Turnbull. 'Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset'. In: *Future Generation Computer Systems* 100 (2019), pp. 779–796.

[170] Bo Dong and Xue Wang. 'Comparison deep learning method to traditional methods using for network intrusion detection'. In: *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*. IEEE. 2016, pp. 581–585.

[171] Peilun Wu and Hui Guo. 'LuNet: A Deep Neural Network for Network Intrusion Detection'. In: *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2019, pp. 617–624.

[172] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, Joarder Kamruzzaman and Ammar Alazab. 'A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks'. In: *Electronics* 8.11 (2019), p. 1210.

[173] Ian J Goodfellow, Jonathon Shlens and Christian Szegedy. 'Explaining and harnessing adversarial examples'. In: *arXiv preprint arXiv:1412.6572* (2014).

[174] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu and Dawn Song. 'Generating adversarial examples with adversarial networks'. In: *arXiv preprint arXiv:1801.02610* (2018).

[175] Zhengli Zhao, Dheeru Dua and Sameer Singh. 'Generating natural adversarial examples'. In: *arXiv preprint arXiv:1710.11342* (2017).

[176] Xiaoyong Yuan, Pan He, Qile Zhu and Xiaolin Li. 'Adversarial examples: Attacks and defenses for deep learning'. In: *IEEE transactions on neural networks and learning systems* 30.9 (2019), pp. 2805–2824.

[177] Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Eleftherios Anastasiadis and George Loukas. 'A taxonomy and survey of attacks against machine learning'. In: *Computer Science Review* 34 (2019), p. 100199.

[178] Panagiotis Kantartopoulos, Nikolaos Pitropakis, Alexios Mylonas and Nicolas Kylilis. 'Exploring Adversarial Attacks and Defences for Fake Twitter Account Detection'. In: *Technologies* 8.4 (2020), p. 64.

[179] Huang Xiao, Battista Biggio, Blaine Nelson, Han Xiao, Claudia Eckert and Fabio Roli. 'Support vector machines under adversarial label contamination'. In: *Neurocomputing* 160 (2015), pp. 53–62.

[180] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein and J Doug Tygar. 'Adversarial machine learning'. In: *Proceedings of the 4th ACM workshop on Security and artificial intelligence*. 2011, pp. 43–58.

[181] Shixiang Gu and Luca Rigazio. 'Towards deep neural network architectures robust to adversarial examples'. In: *arXiv preprint arXiv:1412.5068* (2014).

[182] Sam Grierson, Craig Thomson, Pavlos Papadopoulos and Bill Buchanan. 'Minmax Training: Adversarially Robust Learning Models for Network Intrusion Detection Systems'. In: *2021 14th International Conference on Security of Informa-*

tion and Networks (SIN). Vol. 1. 2021, pp. 1–8. DOI: 10.1109/SIN54109.2021.9699157.

[183] Reza Shokri, Marco Stronati, Congzheng Song and Vitaly Shmatikov. 'Membership inference attacks against machine learning models'. In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 3–18.

[184] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz and Michael Backes. 'Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models'. In: *arXiv preprint arXiv:1806.01246* (2018).

[185] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier and Hervé Jégou. 'White-box vs black-box: Bayes optimal strategies for membership inference'. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5558–5567.

[186] Jason W. Bentley, Daniel Gibney, Gary Hoppenworth and Sumit Kumar Jha. 'Quantifying Membership Inference Vulnerability via Generalization Gap and Other Model Metrics'. In: *arXiv* (2020). arXiv: 2009.05669. URL: http://arxiv.org/abs/2009.05669.

[187] Mohammad Yaghini, Bogdan Kulynych and Carmela Troncoso. 'Disparate Vulnerability: On the unfairness of privacy attacks against machine learning'. In: *arXiv* (2019). arXiv: 1906.00389.

[188] Matt Fredrikson, Somesh Jha and Thomas Ristenpart. 'Model inversion attacks that exploit confidence information and basic countermeasures'. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 2015, pp. 1322–1333.

[189] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page and Thomas Ristenpart. 'Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing'. In: *23rd USENIX Security Symposium (USENIX Security 14)*. 2014, pp. 17–32.

[190] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li and Dawn Song. 'The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks'. In: *arXiv preprint arXiv:1911.07135* (2019).

[191] Boris Delibašić, Milan Vukićević, Miloš Jovanović and Milija Suknović. 'White-Box or Black-Box Decision Tree Algorithms: Which to Use in Education?' In: *IEEE Transactions on Education* 56.3 (2012), pp. 287–291.

[192] Ziqi Yang, Ee Chien Chang and Zhenkai Liang. 'Adversarial neural network inversion via auxiliary knowledge alignment'. In: *arXiv* (2019). ISSN: 23318422. arXiv: 1902.08552.

[193] Seira Hidano, Takao Murakami, Shuichi Katsumata, Shinsaku Kiyomoto and Goichiro Hanaoka. 'Model inversion attacks for online prediction systems: Without knowledge of non-sensitive attributes'. In: *IEICE Transactions on Information and Systems* E101D.11 (2018), pp. 2665–2676. ISSN: 17451361. DOI: 10.1587/transinf.2017ICP0013.

[194] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali and Giovanni Felici. 'Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers'. In: *International Journal of Security and Networks* 10.3 (2015), pp. 137–150. ISSN: 17478413. DOI: 10.1504/IJSN.2015.071829. arXiv: 1306.4447.

[195] Xi Wu, Matthew Fredrikson, Somesh Jha and Jeffrey F. Naughton. 'A methodology for formalizing model-inversion attacks'. In: *Proceedings - IEEE Computer Security Foundations Symposium* 2016-August (2016), pp. 355–370. ISSN: 19401434. DOI: 10.1109/CSF.2016.32.

[196] Congzheng Song, Thomas Ristenpart and Vitaly Shmatikov. 'Machine learning models that remember too much'. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017, pp. 587–601.

[197] Congzheng Song and Reza Shokri. ‘Robust Membership Encoding: Inference Attacks and Copyright Protection for Deep Learning’. In: *arXiv* (2019). arXiv: 1909.12982. URL: http://arxiv.org/abs/1909.12982.

[198] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter and Thomas Ristenpart. ‘Stealing machine learning models via prediction apis’. In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016, pp. 601–618.

[199] Nicholas Carlini and David Wagner. ‘Towards evaluating the robustness of neural networks’. In: *2017 ieee symposium on security and privacy (sp)*. IEEE. 2017, pp. 39–57.

[200] Jianbo Chen, Michael I Jordan and Martin J Wainwright. ‘Hopskipjumpattack: A query-efficient decision-based attack’. In: *arXiv preprint arXiv:1904.02144* 3 (2019).

[201] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik and Ananthram Swami. ‘Practical black-box attacks against machine learning’. In: *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 2017, pp. 506–519.

[202] Yingzhe He, Guozhu Meng, Kai Chen, Xingbo Hu and Jinwen He. ‘Towards Privacy and Security of Deep Learning Systems: A Survey’. In: *arXiv preprint arXiv:1911.12562* (2019).

[203] Zilong Lin, Yong Shi and Zhi Xue. ‘Idsgan: Generative adversarial networks for attack generation against intrusion detection’. In: *arXiv preprint arXiv:1809.02077* (2018).

[204] Jacob Steinhardt, Pang Wei W Koh and Percy S Liang. ‘Certified defenses for data poisoning attacks’. In: *Advances in neural information processing systems*. 2017, pp. 3517–3529.

[205] Ricky Laishram and Vir Virander Phoha. ‘Curie: A method for protecting SVM Classifier from Poisoning Attack’. In: *arXiv preprint arXiv:1606.01584* (2016).

[206] Battista Biggio, Blaine Nelson and Pavel Laskov. ‘Poisoning attacks against support vector machines’. In: *arXiv preprint arXiv:1206.6389* (2012).

[207] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru and Bo Li. ‘Manipulating machine learning: Poisoning attacks and countermeasures for regression learning’. In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 19–35.

[208] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu and Fabio Roli. ‘Towards poisoning of deep learning algorithms with back-gradient optimization’. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 2017, pp. 27–38.

[209] Chaofei Yang, Qing Wu, Hai Li and Yiran Chen. ‘Generative poisoning attack method against neural networks’. In: *arXiv preprint arXiv:1703.01340* (2017).

[210] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, Lingjuan Lyu and Ji Liu. ‘Data poisoning attacks on federated machine learning’. In: *IEEE Internet of Things Journal* (2021).

[211] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson and Tom Goldstein. ‘Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks’. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9389–9398.

[212] Tianyu Gu, Brendan Dolan-Gavitt and Siddharth Garg. ‘Badnets: Identifying vulnerabilities in the machine learning model supply chain’. In: *arXiv preprint arXiv:1708.06733* (2017).

[213] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang and Xiangyu Zhang. ‘Trojaning Attack on Neural Networks’. In: *Department of Computer Science Technical Reports* (2017).

[214] Seda Gürses and Jose M del Alamo. ‘Privacy engineering: Shaping an emerging field of research and practice’. In: *IEEE Security & Privacy* 14.2 (2016), pp. 40–46.

[215] Yod-Samuel Martin and Antonio Kung. ‘Methods and tools for GDPR compliance through privacy and data protection engineering’. In: *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE. 2018, pp. 108–111.

[216] Dwight Horne and Suku Nair. ‘A New Privacy-Enhanced Technology for Fair Matchmaking With Identity Linked Wishes’. In: *IEEE Systems Journal* (2019).

[217] Marcel Gladbach, Ziad Sehili, Thomas Kudrass, Peter Christen and Erhard Rahm. ‘Distributed privacy-preserving record linkage using pivot-based filter techniques’. In: *2018 IEEE 34th International Conference on Data Engineering Workshops (ICDEW)*. IEEE. 2018, pp. 33–38.

[218] Martin Franke, Marcel Gladbach, Ziad Sehili, Florens Rohde and Erhard Rahm. ‘ScaDS research on scalable privacy-preserving record linkage’. In: *Datenbank-Spektrum* 19.1 (2019), pp. 31–40.

[219] Daniel Jurasky and James H Martin. ‘Speech and Language Processing: An introduction to natural language Processing’. In: *Computational Linguistics and Speech Recognition. Prentice Hall, New Jersey* (2000).

[220] Pramod Subramanyan, Rohit Sinha, Ilia Lebedev, Srinivas Devadas and Sanjit A Seshia. ‘A formal foundation for secure remote execution of enclaves’. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017, pp. 2435–2450.

[221] Arup Mondal, Yash More, Ruthu Hulikal Rooparaghunath and Debayan Gupta. ‘Poster: FLATEE: Federated Learning Across Trusted Execution Environments’. In: *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2021, pp. 707–709.

[222] Carlton Shepherd, Raja Naeem Akram and Konstantinos Markantonakis. ‘Remote credential management with mutual attestation for trusted execution environments’. In: *IFIP International Conference on Information Security Theory and Practice*. Springer. 2018, pp. 157–173.

[223] Carlton Shepherd, Ghada Arfaoui, Iakovos Gurulian, Robert P Lee, Konstantinos Markantonakis, Raja Naeem Akram, Damien Sauveron and Emmanuel Conchon. 'Secure and trusted execution: Past, present, and future-a critical review in the context of the internet of things and cyber-physical systems'. In: *2016 IEEE Trustcom/BigDataSE/ISPA* (2016), pp. 168–177.

[224] Caroline Fontaine and Fabien Galand. 'A survey of homomorphic encryption for nonspecialists'. In: *EURASIP Journal on Information Security* 2007.1 (2007), p. 013801.

[225] Sai Sri Sathya, Praneeth Vepakomma, Ramesh Raskar, Ranjan Ramachandra and Santanu Bhattacharya. 'A review of homomorphic encryption libraries for secure computation'. In: *arXiv preprint arXiv:1812.02428* (2018).

[226] Lifang Zhang, Yan Zheng and Raimo Kantoa. 'A review of homomorphic encryption and its applications'. In: *Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications*. 2016, pp. 97–106.

[227] Raphael Bost, Raluca Ada Popa, Stephen Tu and Shafi Goldwasser. 'Machine learning classification over encrypted data.' In: *NDSS*. Vol. 4324. 2015, p. 4325.

[228] Baiyu Li and Daniele Micciancio. 'On the security of homomorphic encryption on approximate numbers'. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2021, pp. 648–677.

[229] Massimo Chenal and Qiang Tang. 'On key recovery attacks against existing somewhat homomorphic encryption schemes'. In: *International Conference on Cryptology and Information Security in Latin America*. Springer. 2014, pp. 239–258.

[230] Oded Goldreich. 'Secure multi-party computation'. In: *Manuscript. Preliminary version* 78 (1998).

[231] Marcella Hastings, Brett Hemenway, Daniel Noble and Steve Zdancewic. 'Sok: General purpose compilers for secure multi-party computation'. In: *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2019, pp. 1220–1237.

[232] Valerie Chen, Valerio Pastro and Mariana Raykova. 'Secure computation for machine learning with SPDZ'. In: *arXiv preprint arXiv:1901.00329* (2019).

[233] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar and Li Zhang. 'Deep learning with differential privacy'. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 308–318.

[234] H Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot and Peter Kairouz. 'A general approach to adding differential privacy to iterative training procedures'. In: *arXiv preprint arXiv:1812.06210* (2018).

[235] Eugene Bagdasaryan, Omid Poursaeed and Vitaly Shmatikov. 'Differential privacy has disparate impact on model accuracy'. In: *Advances in Neural Information Processing Systems*. 2019, pp. 15453–15462.

[236] Nicolas Sartor. *Explaining Differential Privacy in 3 Levels of Difficulty - Aircloak*. `https://aircloak.com/explaining-differential-privacy/`. Accessed on 01 Mar 2022. 2019.

[237] Nicolas Papernot, Martin Abadi, Ulfar Erlingsson, Ian Goodfellow and Kunal Talwar. 'Semi-supervised knowledge transfer for deep learning from private training data'. In: *arXiv preprint arXiv:1610.05755* (2016).

[238] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar and Úlfar Erlingsson. 'Scalable private learning with pate'. In: *arXiv preprint arXiv:1802.08908* (2018).

[239] Tiago AO Alves, Felipe MG França and Sandip Kundu. 'MLPrivacyGuard: Defeating Confidence Information based Model Inversion Attacks on Machine Learning Systems'. In: *Proceedings of the 2019 on Great Lakes Symposium on VLSI*. 2019, pp. 411–415.

[240] Allen B Downey. ‘Evidence for long-tailed distributions in the internet’. In: *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. 2001, pp. 229–241.

[241] Geoffrey Hinton, Oriol Vinyals and Jeff Dean. ‘Distilling the knowledge in a neural network’. In: *arXiv preprint arXiv:1503.02531* (2015).

[242] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha and Ananthram Swami. ‘Distillation as a defense to adversarial perturbations against deep neural networks’. In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, pp. 582–597.

[243] Ji Wang, Weidong Bao, Lichao Sun, Xiaomin Zhu, Bokai Cao and S Yu Philip. ‘Private model compression via knowledge distillation’. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 1190–1197.

[244] Virat Shejwalkar and Amir Houmansadr. ‘Reconciling Utility and Membership Privacy via Knowledge Distillation’. In: *arXiv preprint arXiv:1906.06589* (2019).

[245] Linda Guiga and A. W. Roscoe. *Neural Network Security: Hiding CNN Parameters with GuidedGrad-CAM*. `http://www.cs.ox.ac.uk/files/11814/NNSecurity_new.pdf`. 2020.

[246] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh and Dhruv Batra. ‘Grad-cam: Visual explanations from deep networks via gradient-based localization’. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[247] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh and Patrick McDaniel. ‘Ensemble adversarial training: Attacks and defenses’. In: *arXiv preprint arXiv:1705.07204* (2017).

[248] Jamie Hayes and Olga Ohrimenko. ‘Contamination attacks and mitigation in multi-party machine learning’. In: *Advances in Neural Information Processing Systems*. 2018, pp. 6604–6615.

[249] Nicolas Papernot, Patrick McDaniel and Ian Goodfellow. 'Transferability in machine learning: from phenomena to black-box attacks using adversarial samples'. In: *arXiv preprint arXiv:1605.07277* (2016).

[250] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi and Pascal Frossard. 'Deepfool: a simple and accurate method to fool deep neural networks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2574–2582.

[251] Seong Joon Oh, Mario Fritz and Bernt Schiele. 'Adversarial Image Perturbation for Privacy Protection–A Game Theory Perspective'. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1482–1491.

[252] Milad Nasr, Reza Shokri and Amir Houmansadr. 'Machine learning with membership privacy using adversarial regularization'. In: *Proceedings of the ACM Conference on Computer and Communications Security* (2018), pp. 634–646. ISSN: 15437221. DOI: `10.1145/3243734.3243855`. arXiv: `1807.05852`.

[253] Alex Beutel, Jilin Chen, Zhe Zhao and Ed H Chi. 'Data Decisions and Theoretical Implications when Adversarially Learning Fair Representations'. In: *arXiv* (2017). arXiv: `arXiv:1707.00075v2`.

[254] Seyed Ali Osia, Ali Shahin Shamsabadi, Sina Sajadmanesh, Ali Taheri, Kleomenis Katevas, Hamid R. Rabiee, Nicholas D. Lane and Hamed Haddadi. 'A Hybrid Deep Learning Architecture for Privacy-Preserving Mobile Analytics'. In: *IEEE Internet of Things Journal* 7.5 (2020), pp. 4505–4518. ISSN: 23274662. DOI: `10.1109/JIOT.2020.2967734`. arXiv: `1703.02952`.

[255] Xuanqing Liu, Minhao Cheng, Huan Zhang and Cho-Jui Hsieh. 'Towards robust neural networks via random self-ensemble'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 369–385.

[256] Benjamin IP Rubinstein, Blaine Nelson, Ling Huang, Anthony D Joseph, Shinghon Lau, Satish Rao, Nina Taft and J Doug Tygar. 'Antidote: understanding and

defending against poisoning of anomaly detectors'. In: *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*. 2009, pp. 1–14.

[257] Cong Xie, Oluwasanmi Koyejo and Indranil Gupta. 'Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance'. In: *arXiv preprint arXiv:1805.10032* (2018).

[258] Cong Xie. 'Zeno++: robust asynchronous SGD with arbitrary number of Byzantine workers'. In: *arXiv preprint arXiv:1903.07020* (2019).

[259] Cong Xie, Sanmi Koyejo and Indranil Gupta. 'Practical distributed learning: Secure machine learning with communication-efficient local updates'. In: *arXiv preprint arXiv:1903.06996* (2019).

[260] Gabriela F Cretu, Angelos Stavrou, Michael E Locasto, Salvatore J Stolfo and Angelos D Keromytis. 'Casting out demons: Sanitizing training data for anomaly sensors'. In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE. 2008, pp. 81–95.

[261] Battista Biggio, Igino Corona, Giorgio Fumera, Giorgio Giacinto and Fabio Roli. 'Bagging classifiers for fighting poisoning attacks in adversarial classification tasks'. In: *International workshop on multiple classifier systems*. Springer. 2011, pp. 350–359.

[262] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy and Biplav Srivastava. 'Detecting backdoor attacks on deep neural networks by activation clustering'. In: *arXiv preprint arXiv:1811.03728* (2018).

[263] Kang Liu, Brendan Dolan-Gavitt and Siddharth Garg. 'Fine-pruning: Defending against backdooring attacks on deep neural networks'. In: *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer. 2018, pp. 273–294.

[264] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe and Surya Nepal. 'Strip: A defence against trojan attacks on deep neural networks'. In: *Proceedings of the 35th Annual Computer Security Applications Conference*. 2019, pp. 113–125.

[265] Yang Ming and Tingting Zhang. 'Efficient privacy-preserving access control scheme in electronic health records system'. In: *Sensors* 18.10 (2018), p. 3520.

[266] Muneeb Ali, Ryan Shea, Jude Nelson and Michael J Freedman. 'Blockstack: A new decentralized internet'. In: *Whitepaper, May* (2017).

[267] R Fouchereau and K Rychkov. *Global DNS Threat Report Understanding the Critical Role of DNS in Network Security*. Accessed on 01 Mar 2022. 2019. URL: `https://www.efficientip.com/adlpdrc/`.

[268] Muks Hirani, Sarah Jones and Ben Read. *Global DNS Hijacking Campaign: DNS Record Manipulation at Scale*. Accessed on 01 Aug 2022. 2019. URL: `https://www.mandiant.com/resources/global-dns-hijacking-campaign-dns-record-manipulation-at-scale`.

[269] Hyperledger Fabric. *Private Data*. 2019. URL: `http://hyperledger-fabric.readthedocs.io/en/release-1.4/private-data/private-data.htm`.

[270] Elli Androulaki, Angelo De Caro, Matthias Neugschwandtner and Alessandro Sorniotti. 'Endorsement in Hyperledger Fabric'. In: *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE. 2019, pp. 510–519.

[271] Santosh Chokhani, Warwick Ford, Randy Sabett, Charles Merrill and Stephen Wu. 'RFC 2527: Internet X. 509 public key infrastructure certificate policy and certification practices framework'. In: *Internet Engineering Task Force (IETF), RFC* (1999).

[272] Shubhani Aggarwal and Neeraj Kumar. 'Hyperledger'. In: *Advances in computers*. Vol. 121. Elsevier, 2021, pp. 323–343.

[273] Vikram Dhillon, David Metcalf and Max Hooper. *Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You.* Springer, 2017.

[274] Carl Boettiger. 'An introduction to Docker for reproducible research'. In: *ACM SIGOPS Operating Systems Review* 49.1 (2015), pp. 71–79.

[275] Parth Thakkar, Senthil Nathan and Balaji Viswanathan. 'Performance benchmarking and optimizing hyperledger fabric blockchain platform'. In: *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE. 2018, pp. 264–276.

[276] Nishant Garg. *Apache Kafka.* Packt Publishing Ltd, 2013.

[277] Marcus Brandenburger, Christian Cachin, Rüdiger Kapitza and Alessandro Sorniotti. 'Blockchain and trusted computing: Problems, pitfalls, and a solution for hyperledger fabric'. In: *arXiv preprint arXiv:1805.08541* (2018).

[278] Hyperledger Fabric. *Chaincode for Developers.* Accessed on 01 Mar 2022. 2019. URL: `https://hyperledger-fabric.readthedocs.io/en/release-1.4/chaincode4ade.html`.

[279] Aurelie Bayle, Mirko Koscina, David Manset and Octavio Perez-Kempner. 'When blockchain meets the right to be forgotten: technology versus law in the healthcare industry'. In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE. 2018, pp. 788–792.

[280] *Patrick Breyer v Bundesrepublik Deutschland Case C-582/14 ECLI:EU:C:2016:779.* Accessed on 01 Mar 2022. 2016.

[281] Google. *Google Public DNS.* Accessed on 01 Mar 2022. 2018. URL: `https://developers.google.com/speed/public-dns`.

[282] Cricket Liu and Paul Albitz. *DNS and Bind.* "O'Reilly Media, Inc.", 2006.

[283] Edward Fjellskal. *Gamelinux Passive DNS.* Accessed on 01 Mar 2022. 2011. URL: `https://github.com/gamelinux/passivedns`.

[284] Hyperledger Fabric. *Certificates Github*. Accessed on 01 Mar 2022. 2019. URL: `https://github.com/hyperledger/fabric-ca`.

[285] Yonatan Sompolinsky and Aviv Zohar. 'Secure high-rate transaction processing in bitcoin'. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2015, pp. 507–527.

[286] Imran Bashir. *Mastering blockchain*. Packt Publishing Ltd, 2017.

[287] Christoph Schuba. 'Addressing weaknesses in the domain name system protocol'. In: *Master's thesis, Purdue University, West Lafayette, IN* (1993).

[288] Ermin Sakic and Wolfgang Kellerer. 'Response time and availability study of RAFT consensus in distributed SDN control plane'. In: *IEEE Transactions on Network and Service Management* 15.1 (2017), pp. 304–318.

[289] Eric Piscini, David Dalton and Lory Kehoe. *Blockchain and Cyber Security. Let's Discuss*. Accessed on 01 Mar 2022. 2017. URL: `https://www2.deloitte.com/lu/en/pages/technology/articles/blockchain-and-cybersecurity-lets-discuss.html`.

[290] Bruce Momjian. *PostgreSQL: introduction and concepts*. Vol. 192. Addison-Wesley New York, 2001.

[291] Hyperledger Fabric. *Client Identity Chaincode Library*. Accessed on 01 Mar 2022. 2017. URL: `https://github.com/hyperledger/fabric/blob/release-1.1/core/chaincode/lib/cid/README.md`.

[292] Drummond Reed, Manu Sporny, Dave Longely, Christopher Allen, Markus Sabadello and Ryan Grant. *Decentralized Identifiers (DIDs) v1.0*. Accessed on 01 Mar 2022. Jan. 2020. URL: `https://w3c.github.io/did-core/`.

[293] Manu Sporny, Dave Longely and David Chadwick. *Verifiable Credentials Data Model 1.0*. Tech. rep. Accessed on 01 Mar 2022. W3C, Nov. 2019. URL: `https://w3c.github.io/vc-data-model/`.

[294] Hyperledger. *Hyperledger Aries*. Accessed on 01 Mar 2022. 2019. URL: `https://www.hyperledger.org/projects/aries`.

[295] Joel Weise. ‘Public key infrastructure overview’. In: *Sun BluePrints OnLine, August* (2001), pp. 1–27.

[296] Jan Camenisch, Maria Dubovitskaya, Anja Lehmann, Gregory Neven, Christian Paquin and Franz-Stefan Preiss. ‘Concepts and languages for privacy-preserving attribute-based authentication’. In: *IFIP Working Conference on Policies and Research in Identity Management*. Springer. 2013, pp. 34–52.

[297] Paul Dunphy and Fabien AP Petitcolas. ‘A first look at identity management schemes on the blockchain’. In: *IEEE security & privacy* 16.4 (2018), pp. 20–29.

[298] W3C Credential Community Group. *DID Method Registry*. Tech. rep. Accessed on 01 Mar 2022. 2019. URL: `https://w3c-ccg.github.io/did-method-registry/`.

[299] Daniel Hardman. *Peer DID Method Specification*. Tech. rep. Accessed on 01 Mar 2022. 2019. URL: `https://openssi.github.io/peer-did-method-spec/index.html`.

[300] David L Chaum. ‘Untraceable electronic mail, return addresses, and digital pseudonyms’. In: *Communications of the ACM* 24.2 (1981), pp. 84–90.

[301] Oliver Terbu. *DIF starts DIDComm Working Group*. Accessed on 01 Mar 2022. Decentralized Identity Foundation, 2020. URL: `https://medium.com/decentralized-identity/dif-starts-didcomm-working-group-9c114d9308dc`.

[302] JEREMY Wohlwend. *Elliptic curve cryptography: Pre and post quantum*. Tech. rep. MIT, Tech. Rep, 2016.

[303] Ronald L Rivest, Adi Shamir and Leonard Adleman. ‘A method for obtaining digital signatures and public-key cryptosystems’. In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[304] Taher ElGamal. ‘A public key cryptosystem and a signature scheme based on discrete logarithms’. In: *IEEE transactions on information theory* 31.4 (1985), pp. 469–472.

[305] Dave Longley, Manu Sporny and Christopher Allen. *Linked Data Signatures 1.0.* Tech. rep. Accessed on 01 Mar 2022. 2019. URL: https://w3c-dvcg.github.io/ld-signatures/.

[306] M Jones, J Bradley and N Sakimura. *JSON Web Signatures.* RFC. Accessed on 01 Mar 2022. May 2015. URL: https://tools.ietf.org/html/rfc7515.

[307] Jan Camenisch and Anna Lysyanskaya. 'A signature scheme with efficient protocols'. In: *International Conference on Security in Communication Networks.* Springer. 2002, pp. 268–289.

[308] Jan Camenisch and Anna Lysyanskaya. 'A Signature Scheme with Efficient Protocols'. en. In: *Security in Communication Networks.* Ed. by Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Stelvio Cimato, Giuseppe Persiano and Clemente Galdi. Vol. 2576. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 268–289. DOI: 10.1007/3-540-36413-7_20. URL: http://link.springer.com/10.1007/3-540-36413-7%5C_20.

[309] David Chaum. 'Security without identification: Transaction systems to make big brother obsolete'. In: *Communications of the ACM* 28.10 (1985), pp. 1030–1044.

[310] M Davie, D Gisolfi, D Hardman, J Jordan, D O'Donnell and D Reed. *The Trust Over IP Stack.* RFC 289. Accessed on 01 Mar 2022. Hyperledger, Oct. 2019. URL: https://github.com/hyperledger/aries-rfcs/tree/master/concepts/0289-toip-stack.

[311] Man Ho Au, Patrick P. Tsang, Willy Susilo and Yi Mu. 'Dynamic Universal Accumulators for DDH Groups and Their Application to Attribute-Based Anonymous Credential Systems'. In: *Topics in Cryptology – CT-RSA 2009.* Ed. by Marc Fischlin. Vol. 5473. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 295–308. URL: http://link.springer.com/10.1007/978-3-642-00862-7_20 (visited on 22/08/2019).

[312] Government of British Columbia. *British Columbia's Verifiable Organizations.* Accessed on 01 Mar 2022. 2018. URL: https://orgbook.gov.bc.ca/en/home.

[313] Open Sourcing Mental Illness, LTD. *Mental Health in Tech Survey - Survey on Mental Health in the Tech Workplace in 2014*. Accessed on 01 Mar 2022. 2016. URL: `https://www.kaggle.com/osmi/mental-health-in-tech-survey`.

[314] Takayuki Nishio and Ryo Yonetani. 'Client selection for federated learning with heterogeneous resources in mobile edge'. In: *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE. 2019, pp. 1–7.

[315] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef and Itai Zeitak. 'Overcoming forgetting in federated learning on non-iid data'. In: *arXiv preprint arXiv:1910.07796* (2019).

[316] Kavya Kopparapu and Eric Lin. 'FedFMC: Sequential Efficient Federated Learning on Non-iid Data'. In: *arXiv preprint arXiv:2006.10937* (2020).

[317] Forest Agostinelli, Matthew Hoffman, Peter Sadowski and Pierre Baldi. 'Learning activation functions to improve deep neural networks'. In: *arXiv preprint arXiv:1412.6830* (2014).

[318] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan and Stephen Marshall. 'Activation functions: Comparison of trends in practice and research for deep learning'. In: *arXiv preprint arXiv:1811.03378* (2018).

[319] Qiang Yang, Yang Liu, Tianjian Chen and Yongxin Tong. 'Federated machine learning: Concept and applications'. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–19.

[320] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson and Mats Jirstrand. 'A performance evaluation of federated learning algorithms'. In: *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*. 2018, pp. 1–8.

[321] William J Buchanan, Muhammad Ali Imran, Masood Ur Rehman, Lei Zhang, Qammer H Abbasi, Christos Chrysoulas, David Haynes, Nikolaos Pitropakis and Pavlos Papadopoulos. 'Review and critical analysis of privacy-preserving

infection tracking and contact tracing'. In: *Frontiers in Communications and Networks* 1 (2020), p. 2.

[322] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin and Vitaly Shmatikov. 'How to backdoor federated learning'. In: *arXiv preprint arXiv:1807.00459* (2018).

[323] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal and Seraphin Calo. 'Analyzing federated learning through an adversarial lens'. In: *arXiv preprint arXiv:1811.12470* (2018).

[324] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang and Ji Liu. 'Data Poisoning Attacks on Federated Machine Learning'. In: *arXiv preprint arXiv:2004.10020* (2020).

[325] Piyush Goyal and Anurag Goyal. 'Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark'. In: *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE. 2017, pp. 77–81.

[326] Antony Martin, Simone Raponi, Theo Combe and Roberto Di Pietro. 'Docker ecosystem–vulnerability analysis'. In: *Computer Communications* 122 (2018), pp. 30–43.

[327] GCHQ. *CyberChef - The Cyber Swiss Army Knife*. Accessed on 01 Mar 2022. 2020. URL: `https://gchq.github.io/CyberChef/`.

[328] Bruce Schneier. 'Two-factor authentication: too little, too late'. In: *Communications of the ACM* 48.4 (2005), p. 136.

[329] Felix Lau, Stuart H Rubin, Michael H Smith and Ljiljana Trajkovic. 'Distributed denial of service attacks'. In: *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*. Vol. 3. IEEE. 2000, pp. 2275–2280.

[330] Victor Medel, Omer Rana, Jose Angel Banares and Unai Arronategui. ‘Modelling performance & resource management in kubernetes’. In: *Proceedings of the 9th International Conference on Utility and Cloud Computing*. 2016, pp. 257–262.

[331] OWASP. ‘TOP 10 2017’. In: *The Ten Most Critical Web Application Security Risks. Release Candidate* 2 (2018).

[332] Cynthia Dwork. ‘Differential privacy’. In: *Encyclopedia of Cryptography and Security* (2011), pp. 338–340.

[333] Ilya Mironov. ‘Renyi differential privacy’. In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE. 2017, pp. 263–275.

[334] Yehida Lindell. ‘Secure multiparty computation for privacy preserving data mining’. In: *Encyclopedia of Data Warehousing and Mining*. IGI Global, 2005, pp. 1005–1009.

[335] Will Abramson, Pavlos Papadopoulos, Nikolaos Pitropakis and William J Buchanan. ‘PyDentity: A playground for education and experimentation with the hyperledger verifiable information exchange platform’. In: *Software Impacts* 9 (2021), p. 100101.

[336] Daniele Romanini, Adam James Hall, Pavlos Papadopoulos, Tom Titcombe, Abbas Ismail, Tudor Cebere, Robert Sandmann, Robin Roehm and Michael A Hoeh. ‘Pyvertical: A vertical federated learning framework for multi-headed splitnn’. In: *arXiv preprint arXiv:2104.00489* (2021).

[337] Rajesh Gupta, Sudeep Tanwar, Sudhanshu Tyagi and Neeraj Kumar. ‘Machine learning models for secure data analytics: A taxonomy and threat model’. In: *Computer Communications* 153 (2020), pp. 406–440.

[338] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay and Debdeep Mukhopadhyay. ‘A survey on adversarial attacks and defences’. In: *CAAI Transactions on Intelligence Technology* 6.1 (2021), pp. 25–45.

[339] Yulexis Pacheco. and Weiqing Sun. ‘Adversarial Machine Learning: A Comparative Study on Contemporary Intrusion Detection Datasets’. In: *Proceedings of the 7th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,* INSTICC. SciTePress, 2021, pp. 160–171. ISBN: 978-989-758-491-6. DOI: 10.5220/0010253501600171.

[340] Nicolas Papernot et al. ‘Technical Report on the CleverHans v2.1.0 Adversarial Examples Library’. In: *arXiv preprint arXiv:1610.00768* (2018).

[341] Zheng Wang. ‘Deep learning-based intrusion detection with adversaries’. In: *IEEE Access* 6 (2018), pp. 38367–38384.

[342] Gabriel Fernandez. ‘Deep Learning Approaches for Network Intrusion Detection’. PhD thesis. The University of Texas at San Antonio, 2019.

[343] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow and Rob Fergus. ‘Intriguing properties of neural networks’. In: *arXiv preprint arXiv:1312.6199* (2013).

[344] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand and Victor Lempitsky. ‘Domain-adversarial training of neural networks’. In: *The journal of machine learning research* 17.1 (2016), pp. 2096–2030.

[345] Tom Titcombe, Adam J Hall, Pavlos Papadopoulos and Daniele Romanini. ‘Practical Defences Against Model Inversion Attacks for Split Neural Networks’. In: *arXiv preprint arXiv:2104.05743* (2021).

[346] LLC OpenDNS. ‘PhishTank: An anti-phishing site’. In: *Online: https://www. phishtank. com* (2016). Accessed on 01 Mar 2022.

[347] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir and Banu Diri. ‘Machine learning based phishing detection from URLs’. In: *Expert Systems with Applications* 117 (2019), pp. 345–357.

[348] Samuel Marchal, Jérôme François, Radu State and Thomas Engel. 'Phishstorm: Detecting phishing with streaming analytics'. In: *IEEE Transactions on Network and Service Management* 11.4 (2014), pp. 458–471.

[349] Min-Sheng Lin, Chien-Yi Chiu, Yuh-Jye Lee and Hsing-Kuo Pao. 'Malicious URL filtering—A big data application'. In: *2013 IEEE international conference on big data*. IEEE. 2013, pp. 589–596.

[350] Samuel Marchal, Jérôme François, Cynthia Wagner, Radu State, Alexandre Dulaunoy, Thomas Engel and Olivier Festor. 'DNSSM: A large scale passive DNS security monitoring framework'. In: *2012 IEEE Network Operations and Management Symposium*. IEEE. 2012, pp. 988–993.

[351] Alexa. *The top 1.000.000 sites on the web*. Accessed on 01 Mar 2022. 2019. URL: `https://www.alexa.com/topsites`.

[352] Will Abramson, Nicky Hickman and Nick Spencer. 'Evaluating Trust Assurance in Indy-based Identity Networks using Public Ledger Data'. In: *Frontiers in Blockchain* 4 (2021), p. 18.

[353] Randall Smith. *Docker Orchestration*. Packt Publishing Ltd, 2017.

[354] Hyperledger. *Hyperledger Aries Cloud Agent - Python*. Accessed on 01 Mar 2022. 2019. URL: `https://github.com/hyperledger/aries-cloudagent-python`.

[355] Ngrok. *Ngrok Service*. Accessed on 01 Mar 2022. 2021. URL: `https://ngrok.com/`.

[356] Andrew Trask, Emma Bluemke, Ben Garfinkel, Claudia Ghezzou Cuervas-Mons and Allan Dafoe. 'Beyond Privacy Trade-offs with Structured Transparency'. In: *arXiv preprint arXiv:2012.08347* (2020).

[357] Adam James Hall et al. 'Syft 0.5: A Platform for Universally Deployable Structured Transparency'. In: *arXiv preprint arXiv:2104.12385* (2021).

[358] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson et al. 'Communication-efficient learning of deep networks from decentralized data'. In: *arXiv preprint arXiv:1602.05629* (2016).

[359] Brendan McMahan and Daniel Ramage. 'Federated learning: Collaborative machine learning without centralized training data'. In: *Google Research Blog* 3 (2017). Accessed on 01 Mar 2022. URL: `https://ai.googleblog.com/2017/04/%20federated-learning-collaborative.html`.

[360] Keith Bonawitz et al. 'Towards federated learning at scale: System design'. In: *arXiv preprint arXiv:1902.01046* (2019).

[361] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert and Jonathan Passerat-Palmbach. 'A generic framework for privacy preserving deep learning'. In: *arXiv preprint arXiv:1811.04017* (2018).

[362] Sabine McConnell and David B Skillicorn. 'Building predictors from vertically distributed data'. In: *Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research*. 2004, pp. 150–162.

[363] Nick Angelou et al. 'Asymmetric Private Set Intersection with Applications to Contact Tracing and Private Vertical Federated Machine Learning'. In: *arXiv preprint arXiv:2011.09350* (2020).

[364] Murat Kantarcioglu and Chris Clifton. 'Privacy-preserving distributed mining of association rules on horizontally partitioned data'. In: *IEEE transactions on knowledge and data engineering* 16.9 (2004), pp. 1026–1037.

[365] OpenMined. *PyVertical*. Accessed on 01 Mar 2022. 2020. URL: `https://github.com/OpenMined/PyVertical`.

[366] Li Deng. 'The mnist database of handwritten digit images for machine learning research [best of the web]'. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[367] Nick Angelou, Ayoud Benaissa, Bogdan Cebere, Will Clark, Phillipp Schoppmann, Rutuja Surve, Daniel Liu and Ben Szymbow. *PSI Source Code*. Accessed on 01 Mar 2022. 2020. URL: `https://github.com/OpenMined/PSI`.

[368] Yang Liu, Xiong Zhang and Libin Wang. 'Asymmetrically Vertical Federated Learning'. In: *arXiv preprint arXiv:2004.07427* (2020).

[369] Praneeth Vepakomma, Tristan Swedish, Ramesh Raskar, Otkrist Gupta and Abhimanyu Dubey. 'No Peek: A Survey of private distributed deep learning'. In: *arXiv preprint arXiv:1812.03288* (2018).

[370] Yann LeCun, Léon Bottou, Yoshua Bengio and Patrick Haffner. 'Gradient-based learning applied to document recognition'. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[371] Gregory Cohen, Saeed Afshar, Jonathan Tapson and Andre Van Schaik. 'EMNIST: Extending MNIST to handwritten letters'. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 2921–2926.

[372] Carl Edward Rasmussen. 'Gaussian processes in machine learning'. In: *Summer school on machine learning*. Springer. 2003, pp. 63–71.

Appendix A

List of Additional Publications

A list of further peer-reviewed publications that provided the background related to the context of this thesis can be seen, as follows:

- Buchanan, W. J., Imran, M. A., Rehman, M. U., Zhang, L., Abbasi, Q. H., Chrysoulas, C., Haynes, D., Pitropakis, N., & **Papadopoulos, P.** (2020). Review and critical analysis of privacy-preserving infection tracking and contact tracing. Frontiers in Communications and Networks, 1, 2.

- Christou, O., Pitropakis, N., **Papadopoulos, P.**, McKeown, S. & Buchanan, W. (2020). Phishing URL Detection Through Top-level Domain Analysis: A Descriptive Approach. In Proceedings of the 6th International Conference on Information Systems Security and Privacy - ICISSP, ISBN 978-989-758-399-5 ISSN 2184-4356, pages 289-298. DOI: 10.5220/0008902202890298

- Abramson, W., Hall, A. J., **Papadopoulos, P.**, Pitropakis, N., & Buchanan, W. J. (2020). A Distributed Trust Framework for Privacy-Preserving Machine Learning. In International Conference on Trust and Privacy in Digital Business (pp. 205-220). Springer, Cham.

- Abramson, W., **Papadopoulos, P.**, Pitropakis, N., & Buchanan, W. J. (2021). PyDentity: A playground for education and experimentation with the hyperledger verifiable information exchange platform. Software Impacts, 9, 100101.

- Angelou, N., Benaissa, A., Cebere, B., Clark, W., Hall, A. J., Hoeh, M. A., Liu, D., **Papadopoulos, P.**, Roehm, R., Sandmann, R., Schoppmann, P., & Titcombe, T. (2020). Asymmetric Private Set Intersection with Applications to Contact Tracing and Private Vertical Federated Machine Learning. arXiv preprint arXiv:2011.09350.

- Romanini, D., Hall, A. J., **Papadopoulos, P.**, Titcombe, T., Ismail, A., Cebere, T., Sandmann, R., Roehm, R., & Hoeh, M. A. (2021). PyVertical: A Vertical Federated Learning Framework for Multi-headed SplitNN. arXiv preprint arXiv:2104.00489.

- Titcombe, T., Hall, A. J., **Papadopoulos, P.**, & Romanini, D. (2021). Practical defences against model inversion attacks for split neural networks. arXiv preprint arXiv:2104.05743.

- Young, E. H., Chrysoulas, C., Pitropakis, N., **Papadopoulos, P.**, & Buchanan, W. J. (2021, October). Evaluating Tooling and Methodology when Analysing Bitcoin Mixing Services After Forensic Seizure. In 2021 International Conference on Data Analytics for Business and Industry (ICDABI) (pp. 650-654). IEEE.

- McDonald, G., **Papadopoulos, P.**, Pitropakis, N., Ahmad, J., & Buchanan, W. J. (2022). Ransomware: Analysing the Impact on Windows Active Directory Domain Services. Sensors, 22(3), 953.

- Ali, H., **Papadopoulos, P.**, Ahmad, J., Pitropakis, N., Jaroucheh, Z. and Buchanan, W.J., "Privacy-preserving and Trusted Threat Intelligence Sharing using Distributed Ledgers," 2021 14th International Conference on Security of Information and Networks (SIN), 2021, pp. 1-6, doi: 10.1109/SIN54109.2021.9699366.

- Hughes, K., **Papadopoulos, P.**, Pitropakis, N., Smales, A., Ahmad, J., & Buchanan, W. J. (2021). Browsers' Private Mode: Is It What We Were Promised?. Computers, 10(12), 165.

- Chrysoulas, C., Thomson, A., Pitropakis, N., **Papadopoulos, P.**, Lo, O., Buchanan, W. J., ... & Tsolis, D. (2021). GLASS: Towards Secure and Decentralized eGovernance Services using IPFS. arXiv preprint arXiv:2109.08566.

- Grierson, S., Thomson, C., **Papadopoulos, P.** and Buchanan, B. "Min-max Training: Adversarially Robust Learning Models for Network Intrusion Detection Systems," 2021 14th International Conference on Security of Information and Networks (SIN), 2021, pp. 1-8, doi: 10.1109/SIN54109.2021.9699157.

- Lo, O., Buchanan, W. J., Sayeed, S., **Papadopoulos, P.**, Pitropakis, N., & Chrysoulas, C. (2022). GLASS: A Citizen-Centric Distributed Data-Sharing Model within an e-Governance Architecture. Sensors, 22(6), 2291.

Appendix B

Machine Learning Metrics

The accuracy of a network intrusion detection system can be calculated using the Equation (B.5). Additionally, the standard Machine Learning metrics can be seen as follows:

$$True\,Positive\,Rate = \frac{TP}{(TP+FN)} \tag{B.1}$$

$$True\,Negative\,Rate = \frac{TN}{(TN+FP)} \tag{B.2}$$

$$False\,Positive\,Rate = \frac{FP}{(FP+TN)} \tag{B.3}$$

$$False\,Negative\,Rate = \frac{FN}{(TP+FN)} \tag{B.4}$$

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)} \tag{B.5}$$

$$Precision = \frac{TP}{(TP+FP)} \tag{B.6}$$

$$Recall = \frac{TP}{(TP+FN)} \tag{B.7}$$

$$F1\,Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{B.8}$$

Appendix C

Automated Machine Learning Analysis of Phishing URLs

DNS abuses rise in popularity in the last few years and becoming even more sophisticated than previously. Hence, to defend successfully against adversaries, the defensive countermeasures need to evolve and incorporate multiple promising technologies, such as ML. Additionally, the automation of the attacks detection is a protective measure that can combat adversaries quickly, without relying upon the human factor that often can be tricked [43]. Splunk can be used to create a flexible infrastructure that can take as input data from various sources such as domain lists, perform automated ML and identify potential abuses from a user-friendly interface. This section presents the methodology required to create this implementation, and an illustration of this system can be seen in Figure C.1. Additionally, this section presents the findings of this infrastructure alongside evaluation and discussion about them.

C.1 Datasets

An ML algorithm's prediction quality is considerably associated with the datasets' quality used for its training process. In order for the ML algorithm to correctly classify the output of a query, the training datasets require to be labelled as *malicious* or *benign*, accordingly. A common technique is the usage of multiple allowlists and blocklists to reduce the potential *memorisations* of the training data from the ML algorithm [43].

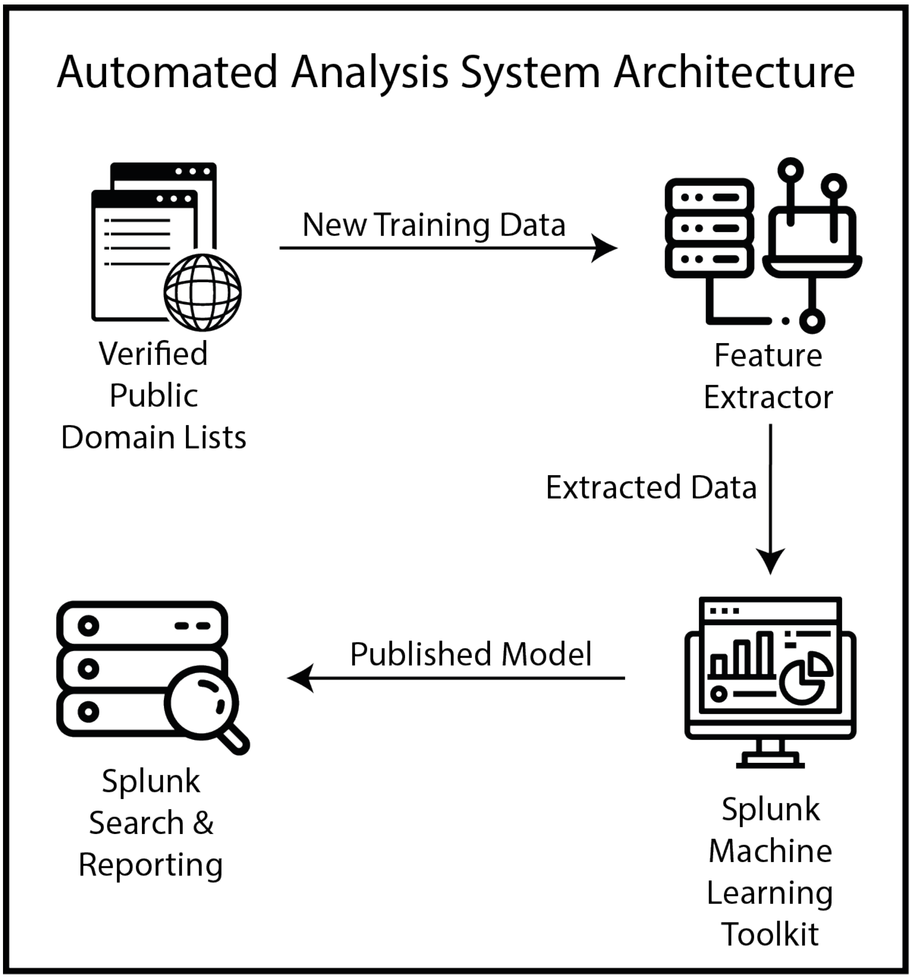

Automated Analysis System Architecture
New Training Data
Verified Public Domain Lists
Feature Extractor
Extracted Data
Published Model
Splunk Search & Reporting
Splunk Machine Learning Toolkit

Figure C.1: System Architecture Diagram [43].

A popular list commonly used by the research community as a benign dataset is the Alexa Top 1 Million domain names database. This database consists of the top 1 million domain name records. However, since these domain names are ordered by popularity, the developers of this database cannot ensure that all the domain names listed are benign and may contain a number of malicious websites. Hence, a countermeasure to combat this issue is the usage of a random percentage of this list and the manual verification of the listed domain names. The process of randomly selecting a percentage of data instead of the entire database can also be used to populate malicious domain name databases such as the Phishtank's blocklist [346], which generally consists of over 400,000 phishing domain name records. The accumulation of data derived from these two lists can be used as a training dataset for the ML algorithm to classify benign or malicious domain names accurately.

Since the manual verification of the Alexa's Top 1M domain name database may still contain a number of malicious records, further tests were conducted using benign and malicious domain name databases from the literature, such as the Sahingoz et al. [347] lists. Furthermore, to conclude the testing precisely and accurately, a third test can be conducted using benign and malicious datasets provided by Marchal et al. [348]. However, the comparison of the findings using combinations of the datasets is critical to ensure the unbiased and accurate prediction of the ML algorithm. Additionally, it should be noted that the feature importance of each dataset combination may change; hence a thorough presentation of the findings is critical.

C.2 Analysis

The ML Toolkit of the Splunk software cannot yet extract the ML features from different datasets. Hence, the usage of other tools commonly used for data pre-processing, such as the Pandas Python library, can help and hasten this process. Since the merged datasets include both benign and malicious domain name records in each test, a new feature should be developed as a new data column that distinguishes the status of each particular domain name record. This domain name status column, or easier the *Type* feature is outlined by the *benign* or *malicious* value, respectively. Furthermore, common Unix tools such as “Regex” can be used to remove redundant content from the dataset. Additionally, since the used datasets may also contain personally identifiable information such as the IP addresses of the client that performed the DNS query or the server that resolved it, any IP addresses should be removed to avoid GDPR legislation issues.

Furthermore, since the benign datasets did not include critical details of a DNS record such as the Time-To-Live, a number of additional features could not be extracted. Consequently, the focus of this work centred on the analysis of the lexical characteristics of the domain name records present in the merged dataset. After the dataset pre-processing, a total of 18 individual features were extracted, divided into two lists descriptive and statistical features, as it can be seen in Table C.1 and Table C.2, respectively. The distinction of the domain name records originates from several variables that can be analysed directly from the domain name records in the case of descriptive features or the several variables that can be analysed after the application of specific mathematical functions to each domain name record in the case of the statistical features. A thorough presentation and explanation of the two lists can be seen in the following sections.

Table C.1: Descriptive features [43].

Feature No.	Feature Description
1	Quantity of URL unique characters
2	Quantity of Domain unique characters
3	Quantity of Suffix unique characters
4	Domain Length
5	Suffix Length
6	Total Length
7	Quantity of Domain Name Numbers
8	Quantity of URL Numbers
9	Quantity of Suffix Numbers
10	Quantity of Symbol Characters in the Domain
11	Quantity of Symbol Characters in the Suffix
12	Total Quantity of Symbol Characters

Table C.2: Statistical features [43].

Feature No.	Feature Description
13	Domain Character Continuity Rate
14	Suffix Character Continuity Rate
15	Shannon Entropy of Domain Name String
16	Shannon Entropy of Suffix String
17	Standard Deviation of the Shannon Entropy of the Two Domain Levels
18	Mean Deviation of the Shannon Entropy of the Two Domain Levels

C.2.1 Descriptive Features

The extraction of 12 features from each domain name record formed the descriptive features list. As presented in the literature, these descriptive features include findings such as that benign domain name records have a lower quantity of numbers or symbols than malicious domain name records. This is because malicious domain name records may be randomly generated through software or because of domain name squatting. Additionally, benign domain name records commonly are shorter than the malicious domain name records [80]. An observation derived from the retrieved malicious datasets is that several malicious domain name records have shorter domain names than subdomains. Hence, even if this is considered unbalanced, the total number of characters may look similar to benign domain name records. Furthermore, since the creators of benign domain name records aim to use easily remembered names, commonly, the quantity of unique characters in a domain name is shorter than malicious domain name records. It should be noted, that often the unique characters in malicious domain name records may be numbers. As such, carefully chosen features were developed to measure the quantity of unique characters, numbers and the length of the domain names, subdomains and suffixes individually.

C.2.2 Statistical Features

The extraction of 6 more features forms the list of the statistical features. As mentioned previously in the descriptive features, benign domain name creators aim to use easily remembered domain names; the price of these domain names tends to be higher than less intuitive domain names. Since the adversaries only use a domain name for a relatively short period of time, they focus on saving as much money as possible. Hence, these less intuitive domain names may contain several similar characters next to each other. This list of statistical features is composed of features such as the presented character continuity rate [349]. To extract this feature, the domain name record is split into tokens of sequential characters according to their nature, such as symbols, numbers or letters. As an example, the "pavlos123-char12" domain name should be split into tokens of sequential characters. Hence, the domain string is split into "pavlos", "123", "-", "char", and "12". In order to calculate the character continuity rate of the domain name, the length of the largest tokens from each category should be added together and then divided by the total length of the domain name. In the previous example, “pavlos” has 6 tokens, “123” has 3 tokens, and “-” has 1 token; hence, the total of these tokens is 10, and divided by the total length of the domain name, which is 16, the character continuity rate infers to 0.625. Another statistical feature that can be extracted from the domain name records is the Shannon entropy of the different domain name levels [350], or their standard and mean deviations.

C.3 Results

As mentioned in the previous sections, the dataset for the ML algorithms contains all the extracted features. Additionally, for the training of the ML algorithms, another data column has been developed that distinguishes if a particular domain name record is benign or malicious. This dataset is used as the input to the Splunk ML Toolkit, which can then be configured to continuously monitor a particular database in order to identify if a domain name record is benign or malicious in real-time with high accuracy. Splunk ML Toolkit is able to split the provided dataset into two subsets, according to the configured split percentage, one intended to train the ML model and one to test it.

Three experiments were developed with different datasets derived from benign and malicious datasets combinations. Test 1 contains 5,000 benign and 5,000 malicious domain name records from the Alexa Top 1M and Phishtank datasets [351, 346]. The merged Test 1 dataset is then 50/50 split, with the first half used for the training of the ML algorithms and the second half for the testing. The Splunk ML Toolkit provides valuable metrics and straightforward hyperparameter configuration to improve the efficiency of any ML algorithm. The chosen algorithms for Test 1 are the Random Forests and SVM to directly compare the findings with other works in the literature. Details about the chosen hyperparameters and the testing performance using the Random Forests and SVM algorithms can be seen in Table C.3 and Table C.4, respectively. For optimal results, careful fine-tuning of the algorithm’s hyperparameters shows that the precision of the Random Forests algorithm can reach up to 89% with 87% recall. The findings of the same experiment but using the SVM algorithm showed that by tweaking its hyperparameters, the precision score can reach up to 89% with 88% recall.

Table C.3: Test 1 Random Forests algorithm performance on the testing dataset [43].

N Estimators	Max Depth	Max Features	Precision	Recall
10	∞	∞	0.87	0.86
10	10	∞	0.89	0.86
10	10	2	0.89	0.87

Table C.4: Test 1 SVM algorithm performance on the testing dataset [43].

C	Gamma	Precision	Recall
1	1/18	0.89	0.87
1	1/50	0.83	0.83
10	1/18	0.90	0.88

For the second experiment, namely Test 2, the dataset includes 70,000 domain name records and derived from Sahingoz et al. [347]. The chosen hyperparameters and the testing performance of the Random Forests and SVM algorithms can be seen in Table C.5 and Table C.6. The maximum performance of the Random Forests algorithm topped at 84% precision with 84% recall, whereas the SVM algorithm's performance was 79% precision with 77% recall after careful optimisation of the hyperparameters.

Table C.5: Test 2 Random Forests algorithm performance on the testing dataset [43].

N Estimators	Max Depth	Max Features	Precision	Recall
10	∞	∞	0.84	0.84
1	∞	∞	0.81	0.81
10	10	∞	0.80	0.80
10	∞	2	0.84	0.84

Table C.6: Test 2 SVM algorithm performance on the testing dataset [43].

C	Gamma	Precision	Recall
1	1/18	0.76	0.76
10	1/18	0.77	0.77
100	1/18	0.78	0.77
100	1/500	0.79	0.77

For the third experiment, Test 3, the dataset consists of 96,000 domain name records and is a combination of the previous dataset with the Phishtorm malicious domain name records dataset [348]. The chosen hyperparameters and testing performance of the tested algorithms, as previously, Random Forests and SVM, can be seen in Table C.7 and Table C.8. The Random Forests algorithm achieved 85% precision with 85% recall, whereas the SVM algorithm reached 81% precision with 81% recall.

Table C.7: Test 3 Random Forests algorithm performance on the testing dataset [43].

N Estimators	Max Depth	Max Features	Precision	Recall
10	∞	∞	0.85	0.85
1	∞	∞	0.83	0.83
10	10	∞	0.83	0.83
10	∞	2	0.85	0.85

Table C.8: Test 2 SVM algorithm performance on the testing dataset [43].

C	Gamma	Precision	Recall
1	1/18	0.79	0.79
100	1/18	0.81	0.81
100	1/100	0.81	0.80
100	1/500	0.80	0.79

The results of the experiments showed that the adjustment of the hyperparameters does not always guarantee an efficiency improvement of the ML algorithms.

C.4 Discussion

As seen in the first experiment, the SVM algorithm outperformed the Random Forests. Additionally, in the case of the SVM algorithm, alteration of the hyperparameters showed significant changes to the results, as seen in TableC.4. Test 1 produced the best results in terms of Precision and Recall, presumably of the lower number of domain name records. The second experiment, namely Test 2, showed that the Random Forests algorithm outperformed the SVM algorithm, as seen in Table C.5 and Table C.6. Additionally, adjustment of the hyperparameters did not reveal any significant perturbation to the results. However, this experiment produced the worse results and is not suited for an automated domain name filtering system. The largest available dataset was used for training for the third experiment, and the results were slightly better than the previous one, as seen in Table C.7 and Table C.8. This experiment proved that the SVM algorithm does not function well when using large datasets, with the Random Forests algorithm's results to prevail.

The experiments confirmed that the selected features were not biased on the used datasets and processed sensible new data. Additionally, it is proved that even if the produced results were not ideal for a fully automated domain name record filtering system, the ML models produced by the experiments could undoubtedly reduce the human input require to manually check, verify and block malicious domain name records and URLs.

C.4.1 Feature Importance

The importance of the chosen set of features is being evaluated using the Splunk ML Toolkit. As it can be seen in Figure C.2, the importance of each feature is visualised in a bar graph. The displayed feature number derives from Table C.1 and Table C.2. In the first experiment, features 1, 6, 16 and 18 held the most weight for the perturbation of the results. That indicates that using the chosen dataset to train the ML model, longer domain name records with more unique characters are often identified as malicious. In the second experiment, the feature importance was more comparable, with features 10, 17 and 18 having more importance than the rest. Finally, for the third experiment, the most crucial features are 6, 10 and 13.

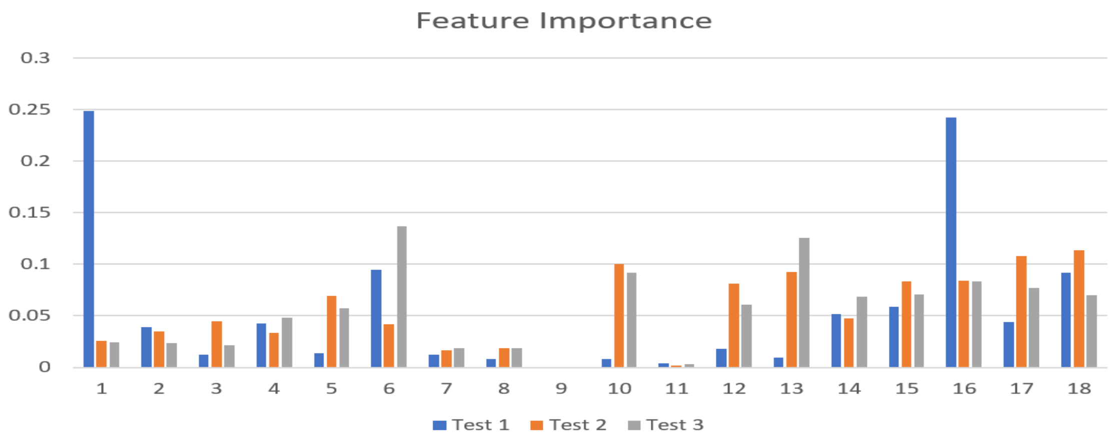


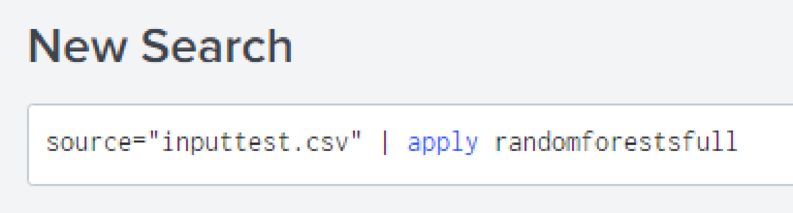
Figure C.2: Feature importance comparison graph [43].

Several interesting observations were made through the presented experiments. Firstly, the feature importance of the second and third experiments showed that larger datasets depending on similar features instead of datasets with fewer data. Secondly, the character continuity rate (features 13 and 14) was the third most crucial feature of Test 3. When this is compared with the literature, it showed that in the work of Lin et al. [349], the character continuity rate feature was the most important. Thirdly, feature 9, which is related to the count of numbers in the domain name record suffix, had no crucial importance in the presented experiments since there were no numbers in any domain name record suffixes.

C.5 Alerting and Further Improvement

After the training of the algorithms, the Splunk ML Toolkit has the functionality to be configured to monitor a specific database continuously in order to predict if a specific domain name record is benign or malicious by utilising the "knowledge" acquired from the presented experiments. Additionally, the toolkit can be configured accordingly to automatically alert the user and be re-trained to continue its learning process, remaining up-to-date with more recent benign and malicious domain name trends and potentially further improve its performance.

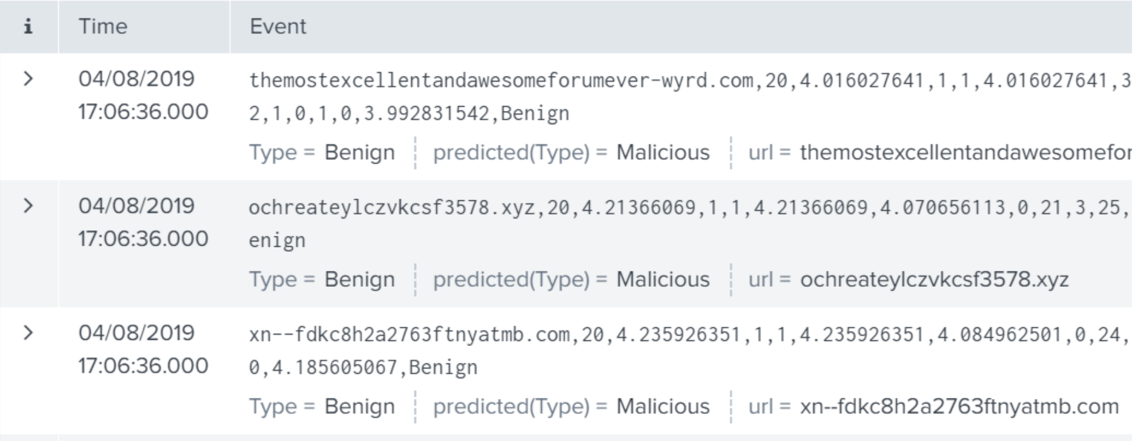
The ML model created by the Splunk ML Toolkit can be “applied” into a specific database, such as a “CSV” file, as it can be seen in Figure C.3. The top 3 results of this Splunk query can be seen in Figure C.4, alongside the prediction if a given domain name is benign or malicious by utilising the knowledge of the previously trained ML model.



New Search

```
source="inputtest.csv" | apply randomforestsfull
```

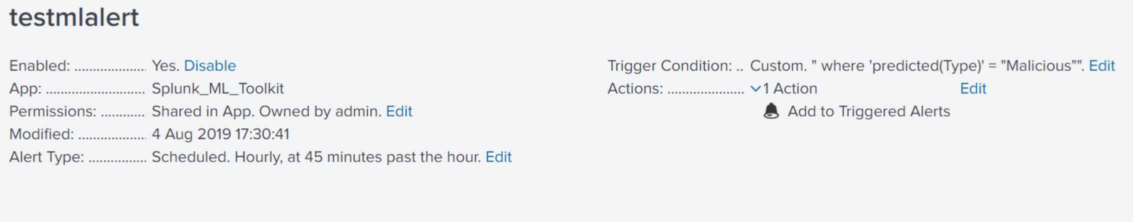
Figure C.3: Applying the knowledge of a machine learning model into a specified database file in Splunk Search [43].



i	Time	Event
>	04/08/2019 17:06:36.000	themostexcellentandawesomeforumever-wyrd.com,20,4.016027641,1,1,4.016027641,3 2,1,0,1,0,3.992831542,Benign Type = Benign ¦ predicted(Type) = Malicious ¦ url = themostexcellentandawesomefor
>	04/08/2019 17:06:36.000	ochreateylczvkcsf3578.xyz,20,4.21366069,1,1,4.21366069,4.070656113,0,21,3,25, enign Type = Benign ¦ predicted(Type) = Malicious ¦ url = ochreateylczvkcsf3578.xyz
>	04/08/2019 17:06:36.000	xn--fdkc8h2a2763ftnyatmb.com,20,4.235926351,1,1,4.235926351,4.084962501,0,24, 0,4.185605067,Benign Type = Benign ¦ predicted(Type) = Malicious ¦ url = xn--fdkc8h2a2763ftnyatmb.com

Figure C.4: Sample results from the Splunk’s search machine learning model fit [43].

The configuration of an alert to the user that is using the Splunk platform can be seen in Figure C.5. Since Splunk is continuously monitoring a specified database in real-time, any additions to the database are automatically checked, utilising the trained ML model and accurately predicting if a domain name entry is benign or malicious. Additionally, after a specified timeframe, the ML model can automatically initiate a re-training using the newly added domain name records. This alerting system is fully automated without requiring any further modifications from its end-users.



testmlalert

Enabled: .... Yes. Disable
App: .... Splunk_ML_Toolkit
Permissions: .... Shared in App. Owned by admin. Edit
Modified: .... 4 Aug 2019 17:30:41
Alert Type: .... Scheduled. Hourly, at 45 minutes past the hour. Edit

Trigger Condition: .. Custom. " where 'predicted(Type)' = "Malicious"". Edit
Actions: .... 1 Action Edit
Add to Triggered Alerts

Figure C.5: Splunk machine learning alert [43].

C.6 Summary

The detection of phishing domain name records is more challenging than identifying botnet traffic because phishing URLs mimic benign domain name records not to look suspicious. Hence, the human input would be further minimised in the future since ML algorithms perform very adequately. Additionally, the automated detection of the phishing domain name records using tools such as Splunk ML Toolkit can offer further protection mechanisms. As presented in the previous sections, Splunk can monitor domain name databases to identify malicious domain name records and URLs in real-time. Additionally, the identified benign and malicious records can be used for further training of the ML models to stay up-to-date to recent trends and probably enhance the accuracy of the algorithms even more.

Appendix D

Additional Technical Experimentation related to Digital Identities and Privacy-Preserving Machine Learning

D.1 Case Study: PyDentity and Aries-Jupyter-Playground

The trusted workflow that presented previously is flexible and can be adapted to any use case and domain. However, its extension to an open-source project, namely PyDentity, and an educational platform, namely Aries-Jupyter-Playground, remove the architectural complexities from their users, enabling them to focus on the development of their use case instead of bothering with low-level architectural details [335].

PyDentity allows effortless and straightforward experimentation with SSI technologies using the Hyperledger Aries. Using Aries-Jupyter-Playground, one is able to create their custom use case by focusing on the high-level domain-specific logic and using the provided low-level libraries to handle the technology's complexities. This domain-specific logic may include the issuing of VC, writing of DIDs to public identities blockchain ledgers, the request and presentations of identity proofs, as well as the exchange of text messages. The platform can be integrated with other commercial and open-source mobile identity wallets by inputting a communication request or scanning a QR code. This platform is demonstrated through Jupyter notebooks to incorporate Python code specifics with supporting text. Additionally, the PyDentity is built using Docker containers that contain all the necessary libraries, frameworks and dependencies required to operate as intended, removing this obstacle from the end-users [335].

D.1.1 Architecture

PyDentity was originally developed as an open-source project within the OpenMined open-source community to demonstrate the usage of SSI technologies. Furthermore, the development of low-level libraries that remove the requirement of knowledge of the underlying technologies created the Aries-Jupyter-Playground educational platform, which is a mixture of three Hyperledger projects, Hyperledger Aries, Hyperledger Indy, and Hyperledger Ursa. Hyperledger Aries handles the domain-specific logic, defines the agent's actions and how they interact with each other [352]. The Hyperledger Indy involves everything related to the identity blockchain ledger, the storage of DIDs that the agents resolve, and their resolution. Finally, the Hyperledger Ursa is the lower layer of the three and includes all the low-level encryption libraries used by the other technologies; it should be noted that the logic behind the end-to-end encrypted DIDComm protocol relies on this layer. The experimental code in PyDentity is written primarily in Python programming language with a few examples in Rust, Go, .NET and Javascript. However, this platform is programming language-agnostic, and it can be incorporated into any SSI system written in any programming language. Similarly, it is blockchain-agnostic, and even if the chosen public identities blockchain ledger is built using Hyperledger Indy, any DID storage and resolution system could be used, even a traditional database such as PostgreSQL [290, 335].

The initiation of the Docker containers for each Hyperledger Aries agent participating in the *playground* is being handled by the Docker Compose [353]. These Docker containers include and are being configured as follows:

- The Docker image of the Aries-Cloudagent-Python (ACA-Py) [354] is being used, which is a Hyperledger Aries instance written in Python programming language and is configurable from an environment file that specifies the Hyperledger Indy network that will be used for the DID interactions; that can be a local Hyperledger Indy instance or a public identities blockchain ledger such as Sovrin.

- A PostgreSQL instance for each agent to store all the identity certificates and cryptographic keys required for their interactions.

- A Jupyter notebook for each agent that includes example tutorials on how the PyDentity works and can be customised with domain-specific logic according to the end-users' use cases.

- Optionally, an Ngrok server [355] can be enabled to expose an HTTP port to the public internet in order to allow its tunnelling to the created application and establish external communications instead of local-only demonstration.

An Aries-Jupyter-Playground examples architecture with two actors, Bob and Alice, can be seen in Figure D.1. The platform allows the further extension of the scenario by adding several other participants without requiring any specific knowledge about the underlying SSI technologies and frameworks. Additionally, since it uses Jupyter notebooks, its further demonstration to other researchers and communities is straightforward through code and text blocks [335].

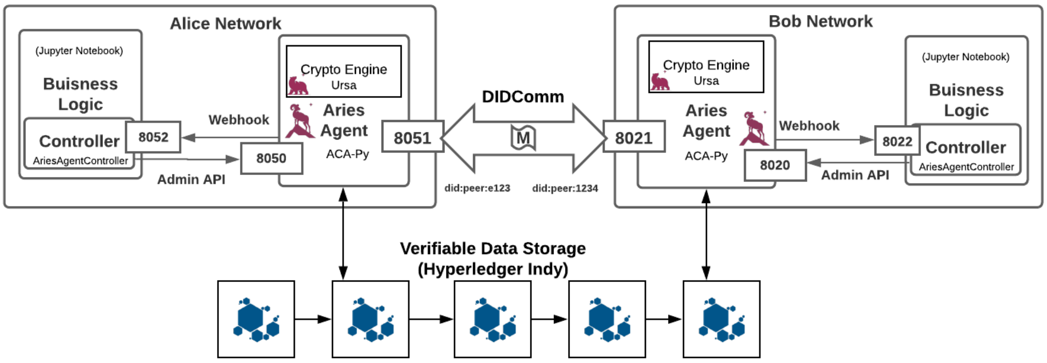


Figure D.1: Aries-Jupyter-Playground Overview [335].

D.1.2 Summary

The PyDentity and its extension to the Aries-Jupyter-Playground are software environments initially created to educate other users about SSI technologies by allowing them to interact with these technologies removing the necessity to master the underlying Hyperledger frameworks. As future work, this system can be extended and be incorporated with other educational platforms related to digital and decentralised identities [356], as well as other projects that would be greatly benefited from this ecosystem such as PyVertical [336] and PySyft [357].

D.2 Federated Machine Learning in Vertically Distributed Datasets

The concept of various FL approaches often involves two types of participants, those who create the ML model and the others who have the necessary sensitive data to train it; hence the term for this concept is called horizontal FL [358, 71, 359, 360, 361]. However, real-world situations are more complex. It is common for *data collectors*, such as various governmental and healthcare departments, services, and banks, to hold data for the same data subjects (the same individuals) [362]. Since the sensitivity of the data, a simple merge of it and share across all the parties is considered a privacy breach and not possible due to regulations. Since the data is split *vertically*, the term vertical ML prevailed [363, 336, 364].

In the previous subsections, the presented FL ecosystem was composed of three hospitals and an ML researcher. The ML researcher created an ML model that was being sent to the three hospitals to train it using their private data. However, the three hospitals shared the same features to train the ML model, and just each one was adding more data samples to the final ML training [68]. If this data was split vertically, such as one hospital held some data about the data subjects (such as the names and surnames of the individuals), another hospital held other data about the same data subjects (such as their mental state), and the third held some others about the same data subjects, then the training of the ML model would not be possible since the researcher that created it could not know which data features each hospital possesses. In this subsection, PyVertical is presented, which is a vertical FL framework [336]. The ML training dataset is split vertically among two data owners and a data scientist. The data partitions are linked using Private Set Intersection (PSI) [363], and the ML model is successfully using Split Neural Networks (SplitNNs) [336].

PyVertical developed as an open-source framework [365] within the OpenMined organisation, and it is the first open-source framework that facilitates vertical FL using SplitNNs [336].

D.2.1 Architecture

In PyVertical, a set of features is split among two data owners and a data scientist. This set of features derives from the MNIST dataset [366] which consists of images of handwritten digits and their labels. The images split vertically, with the left side assigned to a data owner and the right side to the other. Additionally, for the demonstration of the proof-of-concept, the data scientist possesses the set of labels of these images. PyVertical is written in Python programming language, and an overview of its architecture can be seen in Figure D.2. The Data Scientist holds a part of the SplitNN and the labels dataset, whereas the Data Owners hold their images datasets and parts of the SplitNN. This scenario could be extended to a *multi-headed* scenario, which would introduce multiple data owners.

In the vertically split datasets, a number of data samples may intersect between the participants. Hence, a unique ID is associated with each data sample used by the data owners utilising the PSI and mutually agree which data features will be used for the ML training. Furthermore, the data owners sort their datasets according to these IDs and discard non-shared data from their training datasets.

In PyVertical, there is no transmission of the ML model or raw data. The communication of the participants occurs through the PySyft's Duet framework [357] which allows the data scientist to compute data on the data owners' premises remotely.

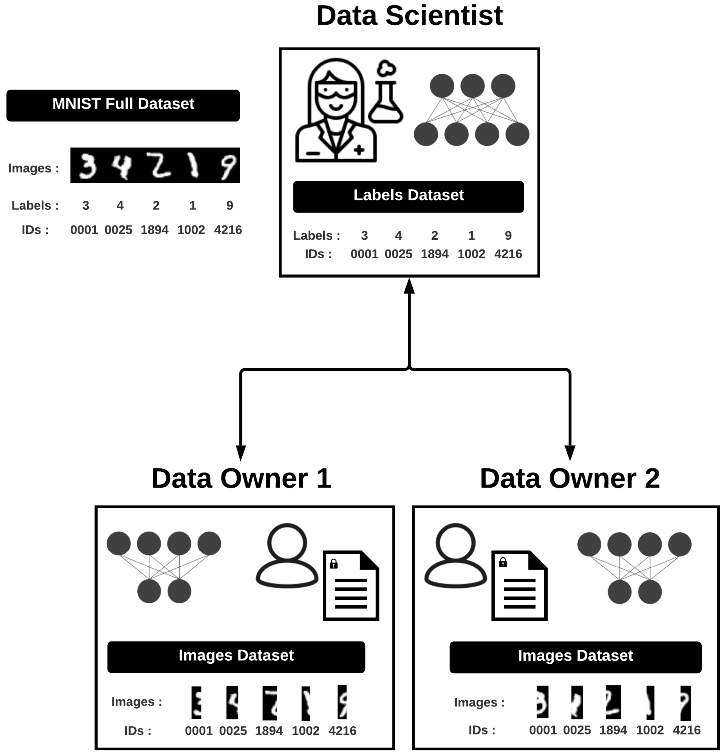


Figure D.2: Parties and datasets in the conducted experiment [336].

D.2.2 Experiment

The objective of the practical experiment is to demonstrate that PyVertical can successfully facilitate federated learning in a vertically distributed dataset. Hence, its objective is not focused on improving the accuracy of the ML algorithm nor carefully tuning its hyperparameters. As mentioned previously, 20.000 images of the MNIST dataset were split into left and right halves and were assigned to the two data owners, whilst their labels were assigned to the data scientist. Furthermore, the linkage of the data samples occurred through the python PSI library [367, 363] in order to investigate all the intersections between the data samples using the previously mentioned unique IDs. Firstly, the data scientist calculates the intersection of their data with one of the data owners. Secondly, the data scientist calculates the intersection with the second data owner; and finally, the data scientist calculates the global intersection of data samples and communicates this information to the data owners. An overview of the usage of PSI can be seen in Figure D.3. In this figure, firstly, the Data Scientist computes the intersection with Data Owner 1. Secondly, the Data Scientist computes the intersection with Data Owner 2. Finally, the Data Scientist computes the global intersection. Additionally, in SplitNNs, each participant trains their own ML model segment; however, the data scientist controls the ML training process and is able to calculate the accuracy and loss of the model using a testing dataset mentioned as the validation dataset, with the results seen in Figure D.4. The technical details of the experiment include the learning rate used for this experiment, which is 0.01 for the data owners' models and 0.1 for the data scientist model using the ReLu activation function and the training epochs that were 30 [336].

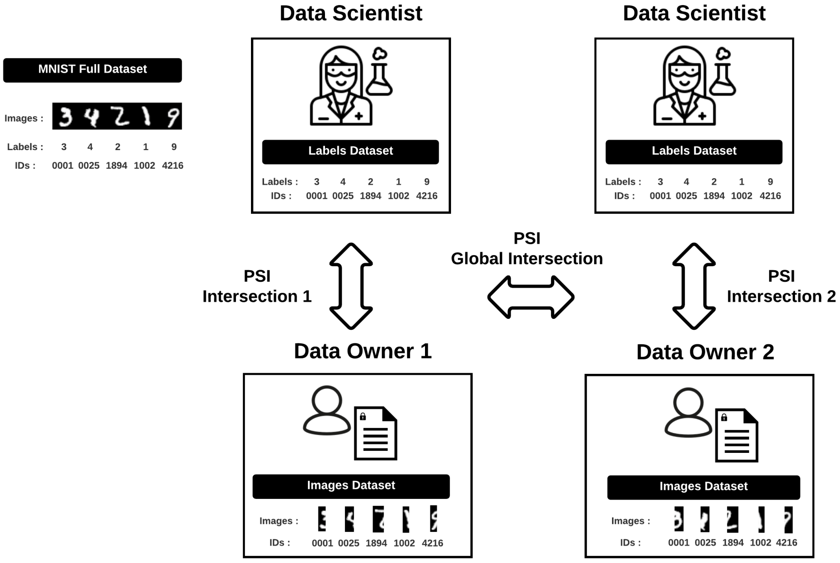


Figure D.3: Computation of intersections [336].

D.2.3 Summary

PyVertical is the first open-source framework that enables the FL on vertically distributed datasets. The experiment presents two data owners that hold the raw data and one data scientist that controls the ML training process. There is no transmission of the ML model nor the raw data throughout this experiment. All the communications occur through the Duet framework that enables the remote computation of data. A future avenue for this project would be its combination with PyDentity to introduce a trust mechanism that also protects all communications through end-to-end encrypted channels [68]. Additionally, the presented experiment was dual-headed, and all the ML model segments had the same size; hence, future work could investigate a multi-headed scenario with multiple data owners and the impact of imbalanced datasets on PyVertical [336, 368].

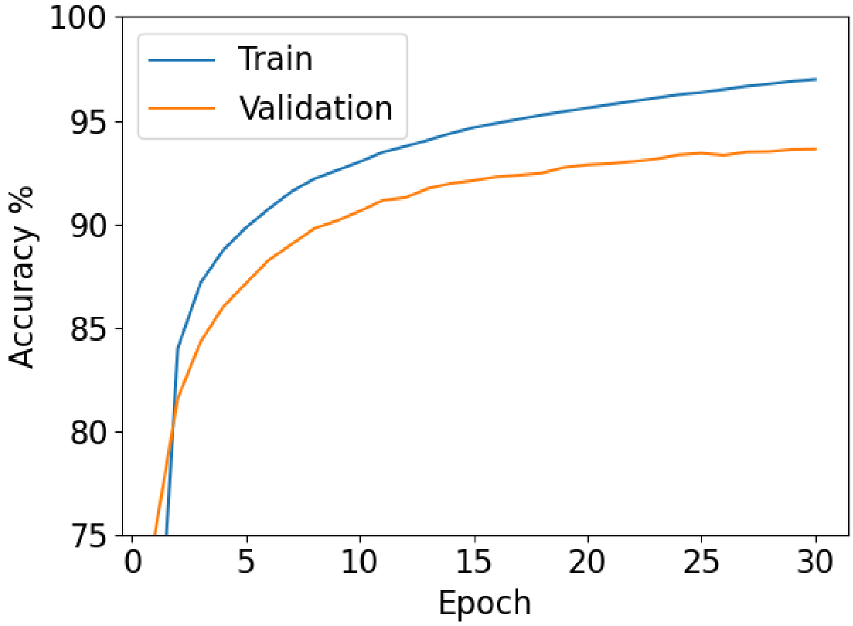


Figure D.4: Train and validation-testing accuracy for an unoptimised dual-headed SplitNN on vertically-partitioned MNIST [336].

Appendix E

Practical Defences Against Model Inversion Attacks for Split Neural Networks

FL tries to solve many of the aforementioned ML attacks since it decentralises the training process. However, it is still susceptible to a number of attacks that, as seen, aim to exploit the ML model itself or reconstruct the data used for training. Similarly, SplitNN approaches add an extra layer of security and privacy; however, they do not fully mitigate all of these attacks. As seen previously, a terrifying ML attack is the model inversion since it aims to exploit the ML model and reconstruct the data used on it. Often data used to train ML models is sensitive, and their security and privacy must be preserved. In this section, a practical defence against model inversion attacks is presented in a SplitNN scenario. Without applying this practical defence, an adversary is able to reconstruct a set of inference-time data with limited knowledge of the data distribution. The presented defence adds noise during the intermediate layers of the ML model's training and successfully mitigates this type of model inversion attacks, as seen in the following subsections [345].

The presented practical defence is the extension of NoPeekNN. NoPeekNN aims to limit the data reconstruction in SplitNN environments, by minimising the distance correlation of the raw input data with data produced during the ML during intermediate representations [54, 369]. NoPeekNN optimises the ML model by combining its loss with the distance correlation loss in order to masquerade the similarity between the intermediate data from the raw input data. NoPeekNN utilises a hyperparameter $\alpha \in [0,\infty]$ to control the perturbation of the loss, and even if it has been shown that it protects the reconstruction of input data efficiently, this method has not been applied to model inversion attacks. As seen in the following subsections, an experimental evaluation compared the presented practical defence with NoPeekNN and concluded to a number of valuable findings [345].

E.1 Architecture

As seen previously, in SplitNNs, each participant controls and trains a model's segment which is mapped to an intermediate ML model representation. The threat model of this experiment considered an adversary that also controls a model segment and accumulates a set of raw data and intermediate data produced by at least one other model segment. The adversary could be a malicious data scientist or a malicious data owner inferring in the training process of another legitimate data owner. The adversary creates their own attacking ML model utilising this set to convert the intermediate data back into raw data inputs. The investigated model inversion attack aims to reconstruct data during the inference of the model and not the reconstruction of training data. Additionally, since the adversary creates their own attacking ML model and does not have any white-box knowledge about the initial ML model, this investigation falls under the black-box model inversion [345].

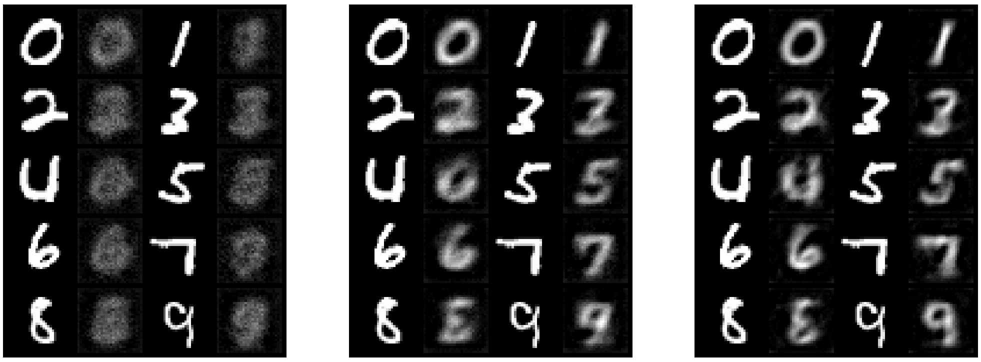
The presented *Noise Defence* adds Laplacian noise after each data owner's ML training of their own model segment before the updates are sent to the data scientist. Alternatively, this noise can also be added during the model segment's ML training. The result of adding noise in the middle of the training process allows the model to adapt to it and improve its utility. Additionally, noise obfuscates the data transmitted to the data scientist through the different model segments and does not allow a malicious data scientist to understand their correlation with the raw input data. The noise defence concurrently with NoPeekNN can be seen in Algorithm 4. The reason for this defence stems from the fact that data owners may not fully trust the data scientists, and they want to preserve the security and privacy of their sensitive data [345].

Algorithm 4 NoPeekNN method with Noise Defence [345, 369].

```
laplacian noise scale $b$
NoPeekNN weight $\alpha$
Data owner model $f_1$ with weights $\theta_1$
Computational server model $f_2$ with weights $\theta_2$
Learning rates $\lambda_1, \lambda_2$
for $epoch \leftarrow 1, 2, \dots, N$ do
    for $inputs, targets \leftarrow dataset$ do
        $intermediate \leftarrow f_1(inputs)$
        $noise \sim L(0, b)$
        $intermediate \leftarrow intermediate + noise$
        $outputs \leftarrow f_2(intermediate)$
        $\theta_1 \leftarrow \theta_1 + \lambda_1 \frac{\partial}{\partial \theta_1} \alpha \mathscr{L}_{dcor}(inputs, intermediate) + \mathscr{L}_{task}(outputs, targets)$
        $\theta_2 \leftarrow \theta_2 + \lambda_2 \frac{\partial}{\partial \theta_2} \mathscr{L}_{task}(outputs, targets)$
    end for
end for
```

For this experiment, the MNIST dataset [370] utilised. More specifically, the experiment is split into two parts; in the first, the adversary has access to some data of the same distribution, and in the second, the adversary derives data to attack from the EMNIST dataset [371]. Regarding the MNIST dataset, from the total of 60,000 images, 40,000 images were used for the training of the ML model and 10,000 for validating purposes, and finally, the remaining 10,000 images were used to train and evaluate the adversary model, split in half for each of those processes. The ML training occurred through 10 epochs, with a 32 batch size and 0.001 learning rate. The hyperparameter α utilised in the noise defence similarly to NoPeekNN, with the values of α set to 0.1, 0.5, 1.0 for the NoPeekNN, and the values of b set to 0.1, 0.5, 1.0, respectively for the presented noise defence. For the second part of the experiment, the impact of the adversary's knowledge on the data distribution has been experimentally evaluated by assigning data from the EMNIST dataset to the adversary [345].

As it can be seen in Figure E.1, the original ML model without any defence applied to it is vulnerable to model inversion reconstructions. The figures represent the reconstruction of images extracted from a classifier with no defence mechanisms applied. Additionally, the figures show one example for each class of the MNIST dataset. Columns 1 and 3 are real datapoints; columns 2 and 4 are reconstructions [345]. More specifically, in Figure E.1b, when 250 MNIST images and more are used, the ground truth of each data point can be reconstructed with high success. In Figure E.1c, the reconstructions are more visible, especially for some particular classes that were not distinctly visible when using 250 MNIST images. This is particularly threatening in the case of highly sensitive data such as medical records [345].



(a) Data reconstructions from an attacker trained on 100 images.

(b) Data reconstructions from an attacker trained on 250 images.

(c) Data reconstructions from an attacker trained on 1250 images.

Figure E.1: MNIST images R reconstructions from adversary models trained on a different numbers of datapoints [345].

E.2 Experimental Evaluation

The presented noise defence, as well as the NoPeekNN, do not significantly impact the accuracy of the ML model. The accuracy of a ML model with the noise defence and NoPeekNN can be seen in Figure E.2. As it can be observed, as the NoPeekNN perturbation increases, the accuracy of the model decreases. However, this impact is not significant since even in high perturbation levels, the accuracy-privacy trade-off is optimal, with shattered reconstructions and an acceptable accuracy score [345].

The accuracy of the ML models and the average distance correlation between intermediate and input data can be seen in Table E.1. Since NoPeekNN is optimised to minimise this correlation, a higher perturbation from it would minimise it, as expected. An interesting finding is that additive noise may partly degrade the NoPeekNN defence since there is a correlation between the noise level increase and the average distance correlation [345].

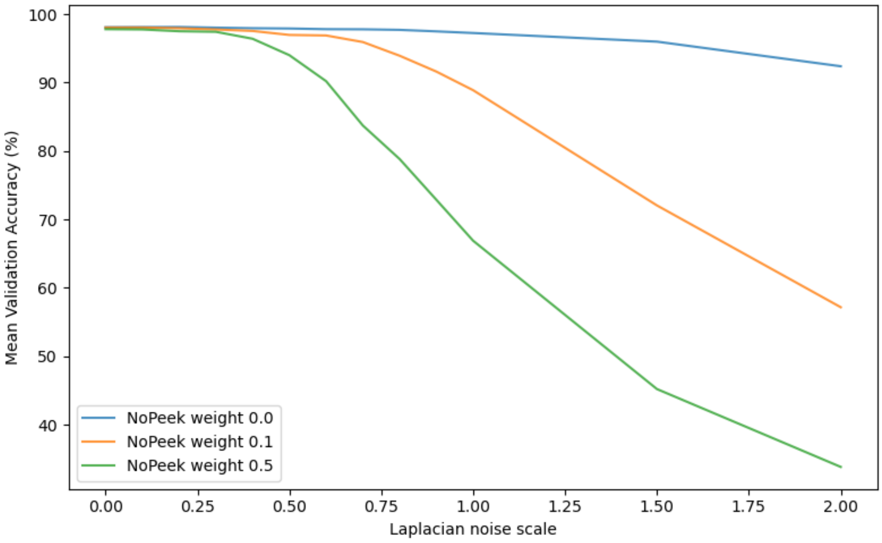


Figure E.2: Accuracies on a validation dataset by classifiers with as a function of the scale of laplacian noise added to intermediate data after training [345].

Noise scale	NoPeekNN weight	Accuracy (%)	Distance Correlation
0.0	0.1	98.04	0.472 ± 0.004
0.0	0.2	97.80	0.390 ± 0.003
0.0	0.5	97.90	0.368 ± 0.004
0.1	0.0	98.19	0.804 ± 0.004
0.1	0.1	97.84	0.474 ± 0.003
0.1	0.5	98.00	0.411 ± 0.004
0.2	0.0	98.00	0.811 ± 0.004
0.2	0.1	97.98	0.491 ± 0.003
0.2	0.5	97.46	0.411 ± 0.003
0.5	0.0	98.24	0.795 ± 0.004
0.5	0.1	97.52	0.525 ± 0.003
0.5	0.5	97.38	0.437 ± 0.003

Table E.1: Validation accuracy and distance correlation between input data and intermediate tensor of classifiers using NoPeekNN and training noise defences [345].

For the second part of the experiments, as seen previously, it was assumed that the adversary utilises the EMNIT dataset to reconstruct images on the ML model that trained on the MNIST dataset. As it can be seen in Figure E.3b, the adversary utilised 5,000 EMNIST images; however, as a result, even if the reconstructions are more fuzzy and shattered than the previous part of the experiment when using MNIST attacking images, the ground truth classes can still be identified, with a comparison presented in Figure E.3a. This finding demonstrates that even if adversaries have partial knowledge about the data distribution of the ML model that they try to attack, a model inversion attack can be alarmingly successful [345]. To further enhance the trust and security of the ML training process, an identity system can be incorporated, as seen in the previous chapter, to transmit the ML models end-to-end encrypted only between trusted participants [68, 47].

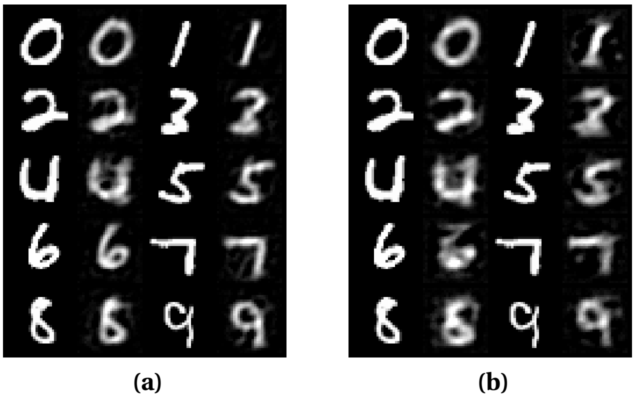
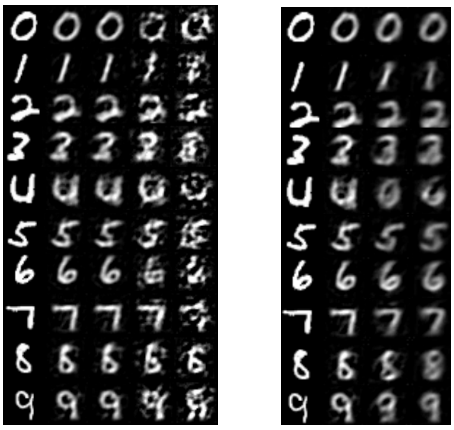


Figure E.3: (**a**) MNIST data reconstructions from an attacker trained on 5,000 MNIST images. (**b**) MNIST data reconstructions from an attacker trained on 5,000 EMNIST images [345].

The image reconstructions using the model inversion attack on varying values of noise are presented in Figure E.4. At a noise value of 1.0, the image reconstructions are entirely shattered. At a noise value of 0.5, large chunks of the reconstructed images are shattered, whilst lower noise values obfuscate only slightly the reconstructions. On the other hand, using NoPeekNN, the image reconstructions obfuscate the most noticeable characteristics of classes, but their general structure remains visible. Similarly to noise defence, lower noise levels do not efficiently protect the model against reconstructions [345]. A combination of noise defence and NoPeekNN can be a feasible and more robust defensive mechanism. Depending on the use case, a ML model may elaborate on the benefits of one or the other [345].



(a)

(b)

Figure E.4: (**a**) Model inversion attack on an MNIST classifier using the *noise defence* mechanism. Left-to-right: true image, reconstructions on models with (0, 0.1, 0.5, 1.0) noise scale. (**b**) Model inversion attack on an MNIST classifier using NoPeekNN defence. Left-to-right: true image, reconstruction on models with (0, 0.1, 0.5) NoPeekNN weighting [345].

E.3 Summary

The noise defence presented in this section extended the NoPeekNN technique by including additive Laplacian noise during the ML model's training. It is able to defend effectively against model inversion attacks with a satisfactory privacy/utility trade-off and can be applied to any SplitNN architecture. Several examples of reconstructed images are presented in order to evaluate the experiment, alongside valuable findings. Additionally, noise defence can be considered comparable to DP. Future avenues for this work would be investigating the impact of noise defence utilising more complex datasets, the calculation of the privacy guarantees it offers, as well as investigating the impact of other noise techniques such as Gaussian noise [345, 372].