University of Sheffield

Developing an Efficient and Privacy-Preserving Energy Theft Detection System for Smart Grids

Arwa Alromih

Supervisors: Prof. John A. Clark & Dr. Prosanta Gope

PhD Thesis

University of Sheffield

Department of Computer Science

September 2023

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. Work submitted for this research degree at the University of Sheffield has not formed part of any other degree either at the University of Sheffield or at another establishment. All sources are acknowledged as references.

Certain parts of the material presented within this thesis have appeared in published or submitted papers elsewhere. Specifically, these are:

- Arwa Alromih, John A. Clark, and Prosanta Gope. "Electricity Theft Detection in the Presence of Prosumers Using a Cluster-based Multi-feature Detection Model". Published in the 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pp. 339-345. IEEE, 2021.
- Arwa Alromih, John A. Clark, and Prosanta Gope. "Privacy-Aware Split Learning Based Energy Theft Detection for Smart Grids". Published in the 24th International Conference on Information and Communications Security (ICICS 2022), pp. 281–300, Springer-Verlag, 2022.
- Arwa Alromih, John A. Clark, and Prosanta Gope. "A Privacy-Preserving Energy Theft Detection Model for Effective Demand-Response Management in Smart Grids". Submitted for publication and a preprint version can be found in arXiv.

Name: Arwa Alromih

Signature: 

Date: 18-09-2023

Acknowledgment

First and foremost, I thank Allah Almighty for giving me the inspiration, patience, time, and strength to finish this work. With Allah's will and mercy, I have been able to achieve all of this.

I cannot express enough thanks and appreciation to my supervisors, Professor John A. Clark and Dr. Prosanta Gope, for their direction, supervision, as well as their comments and guidance on this thesis. Without their considerable efforts, this thesis could not have been carried out to completion. I also offer my sincere appreciation to my supervisory team members, Dr. Neil Walkinshaw and Dr. Andrei Popescu for their fruitful discussions and insightful advice throughout my research work. I would also like to thank my friends and colleagues in the Security of Advanced Systems Research Group at the University of Sheffield for making my PhD journey such an enjoyable experience.

I would like to thank King Saud University and the Saudi Ministry of Education as represented by the Saudi Arabia Cultural Bureau in London for supporting my PhD studies.

I gratefully thank my supportive and encouraging husband, Muath, who has been by my side throughout every stage of my postgraduate studies, living every minute of it, and without whom, I would not have had the courage to start this journey in the first place. I cannot forget to thank my beloved son, Feras, whose smile can always stimulate my ability to work smarter. Likewise, thank you to my beautiful baby girl Sara, for being such a good little baby for the past few months, and making it possible for me to complete what I started.

Finally, but in no way least, my gratitude goes to my mother, brothers and sister for all their love and support throughout my life. I will be eternally grateful to my mother for her unconditional love and encouragement all the way. This thesis is dedicated to her.

Abstract

Energy plays an essential role in our lives. Merging the existing electricity networks with distributed energy resources and information and communications technology (ICT) changes how companies and customers generate, distribute, and consume energy. This integration transforms the legacy electricity networks into smart systems, or what is currently known as the Smart Grid (SG). Smart grid infrastructure has been one of the major industrial revolutions that has attracted widespread adoption across the globe. Therefore, they can be the target of major security risks as they are not inherently secure. In this sector, sensors' and meters' data are the main factors in any decision-making process. This introduces the need to develop appropriate security mechanisms that ensure data integrity. One of the main attacks against data integrity in energy networks is energy theft. This attack can be made by injecting false consumption data into the network. The consequences of a successful energy theft attack on smart grid systems can be severe and far-reaching as it can result in power outages and physical damage to equipment which can be a safety hazard to individuals. Therefore, secure techniques are needed to detect any anomalies or injection attempts and smart meter data integrity should be considered and ensured.

In this thesis, we propose three machine learning (ML) based energy theft detectors that address the existing challenges facing current research in this domain. In particular, we consider the impact proposed by prosumers in launching new types of energy thefts and how to detect them. We also show how to fully utilise data from multiple sources for better detection performance. To decrease the probability of any privacy breaches caused by the use of customers' data, privacy-preserving approaches are proposed. Lastly, we tackle the significant impact on demand management caused by energy thefts by proposing a combined energy theft detector with demand management. The findings presented in this thesis show that our approaches can accurately detect energy thefts, with minimal information leakage. Moreover, the results are also promising in providing a clear link between reliably managing demand when energy theft is considered.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The concept of smart grid (SG) refers to the modernisation of traditional electricity grids that allows dynamic optimisation of operations and maintains a reliable and secure electricity infrastructure. SG uses an advanced metering infrastructure (AMI) that utilises digital information and communication technology (ICT). ICT enables real-time power demand measurements to be exchanged between all components. These high-resolution electricity data, provided by modern smart meters, help balance supply and demand better. New smart grids have also allowed the integration of renewable energy resources at residential levels, enabling consumers to produce energy and sell it to the grid. Although these advancements in smart grid technology and its integration with ICT have brought many advantages, they have also opened up the system to several vulnerabilities. This is because when ICT was first integrated into energy networks, defending against intrusions was not a priority. Smart grids are vulnerable to various types of attacks, including attacks against data integrity, confidentiality and availability.

One of the main attacks against data integrity in energy systems is energy theft. This attack involves manipulating smart meters' fine-grained data through the network. Energy thefts are one of the major causes of non-technical losses (NTL) during electricity transmission and distribution [1]. They are defined as any illegal energy use that violates contract terms. This can be achieved through physical means, such as using a bypass cable (a shunt), or through digital manipulation of meter readings. This can lead to paying nothing (or less) for consumed power or getting paid more for selling to the grid [2]. Globally, energy thefts are the greatest cause of financial losses in the energy market, and it has been reported that around $1 to $6 billion dollars are lost yearly in the UK and the US combined due to these attacks [3].

Protecting smart grid systems against energy thefts is a serious problem, and resilience against them is a prerequisite for reliable operation in energy applications. Our main direction in this thesis is to employ machine learning (ML) techniques to detect these attacks reliably. Here, we develop detection schemes that can detect attacks under various theft situations and can simultaneously ensure data privacy without introducing high communication and communication overheads.

1.1 Aim and Objectives

Energy thefts are one of the most costly attacks launched against smart grids. Hence, they cause significant concerns for both providers and consumers. They are often hard to detect, specifically as they can be launched in different forms. Recently, global electricity consumption has become a burden on energy utilities. This has pushed power systems operators to introduce more efficient and flexible ways for sustainable energy, enabling some energy consumers to engage in energy production. Such customers have become known as "prosumers" (agents that both produce and consume energy). However, introducing those prosumers has allowed energy thefts to be launched on either side (consumption and production). Moreover, advances in smart grid systems have allowed smart metering data to be used in all sorts of energy management, including the accurate detection of thefts. However, they have also raised new challenges concerning how data can be transferred and processed without violating customers' privacy.

Different detection approaches have been proposed in the literature to detect existing energy theft attacks in smart grids (as discussed in Chapter 2). Machine learning-based techniques are widely used as they offer several advantages over other techniques, making them a valuable tool for identifying and combating this problem. However, existing detection approaches address only certain types of consumers' energy thefts and cannot detect thefts by multiple agents or by prosumers. Moreover, we have noticed that previous studies have used either a generalised detection model for all users or a user-specific one. Therefore, opportunities for using data features from different sources can help to find a balance between the two approaches. This is done by grouping users into clusters and creating a reference model for each cluster. This thesis investigates whether having a cluster-based energy theft detection using ML-based approaches is able to detect different energy theft attacks accurately. Hence, our *first hypothesis* is:

Hypothesis 1: *Combining machine learning techniques (clustering and classification) can enhance the detection of a range of thefts, including prosumers thefts.*

The use of machine learning often requires that data is used in its raw form. However, due to privacy concerns and some legal constraints (e.g., GDPR in the EU and CCPA in California), the use of customers' energy data is subject to strict regulations. These privacy policies must be followed to ensure data privacy at all times. For realistic prospects of deployment in real systems, an energy theft detection approach based on machine learning must be privacy-preserving and avoid using data in its raw form. This leads us to formulate our *second hypothesis*:

Hypothesis 2: *A privacy-preserving ML technique that suits the smart grid environment can be developed to accurately and effectively detect energy theft while preserving the privacy of customers' data.*

Most smart grid operations rely on the availability and integrity of smart meters' readings, and any manipulation of these readings can affect operational reliability. Degrading the integrity of such readings may, for example, affect future demand forecasting, which can, in turn, cause disruption to the energy supply. This may lead to outages if a system cannot provide adequate supply or even become an operational safety issue. Thus, an energy theft detection model should be equipped with post-detection mechanisms that enhance demand-response management. We argue that a multi-output neural network offers a particularly appropriate approach to doing this effectively and efficiently. Indeed, we will show that such an approach can provide both detection and prediction functions in the same network. Therefore, our final and *third hypothesis* is:

Hypothesis 3: *A multi-output neural network framework can be used to simultaneously predict the presence of theft, predict its magnitude, and use that estimation to make more accurate forecasts.*

Upon developing these three hypotheses, our primary aim of ensuring the precise and effective detection of various forms of energy theft while preserving customers' privacy will be investigated.

1.2 Thesis Contributions

The major contributions of this thesis are as follows:

- The proposal and evaluation of a cluster-based theft detection method that detects thefts by both consumers and prosumers. The proposed method can detect thefts from new users without the need for historical data.
- The introduction of new electricity theft scenarios, which we term as *balance attacks*. These attacks can balance the amount of electricity stolen from one meter with manipulated values returned from neighbouring meters. This scenario can be difficult to detect with existing detection models.
- The production of a dataset that includes both prosumers' and consumers' profiles.
- The proposal and evaluation of a privacy-preserving energy theft detection approach. The proposed detection is based on a newly proposed variant of split learning, called Three Tier Split Learning, that suits the nature of smart grid infrastructure.
- The proposal of an energy theft detection approach that not only detects energy thefts but also takes post-detection actions that help estimate future demand. The proposed approach is a privacy-preserving scheme that detects energy thefts and estimates the amount of stolen energy, which is then used to manage future demand, even in cases of theft.
- The introduction of quantitative analysis metrics to analyse the privacy of an energy theft detection model using feature inference attacks and distance correlation.

1.3 Structure of the Thesis

This thesis is conceptually structured as illustrated in Figure 1.1 in order to address the aim, objectives and the three hypotheses presented earlier. This includes the following chapters:

In **Chapter 2, Background and Literature Survey**, we outline relevant research topics in the literature, critically survey the existing work and highlight the gaps that led to our hypotheses.

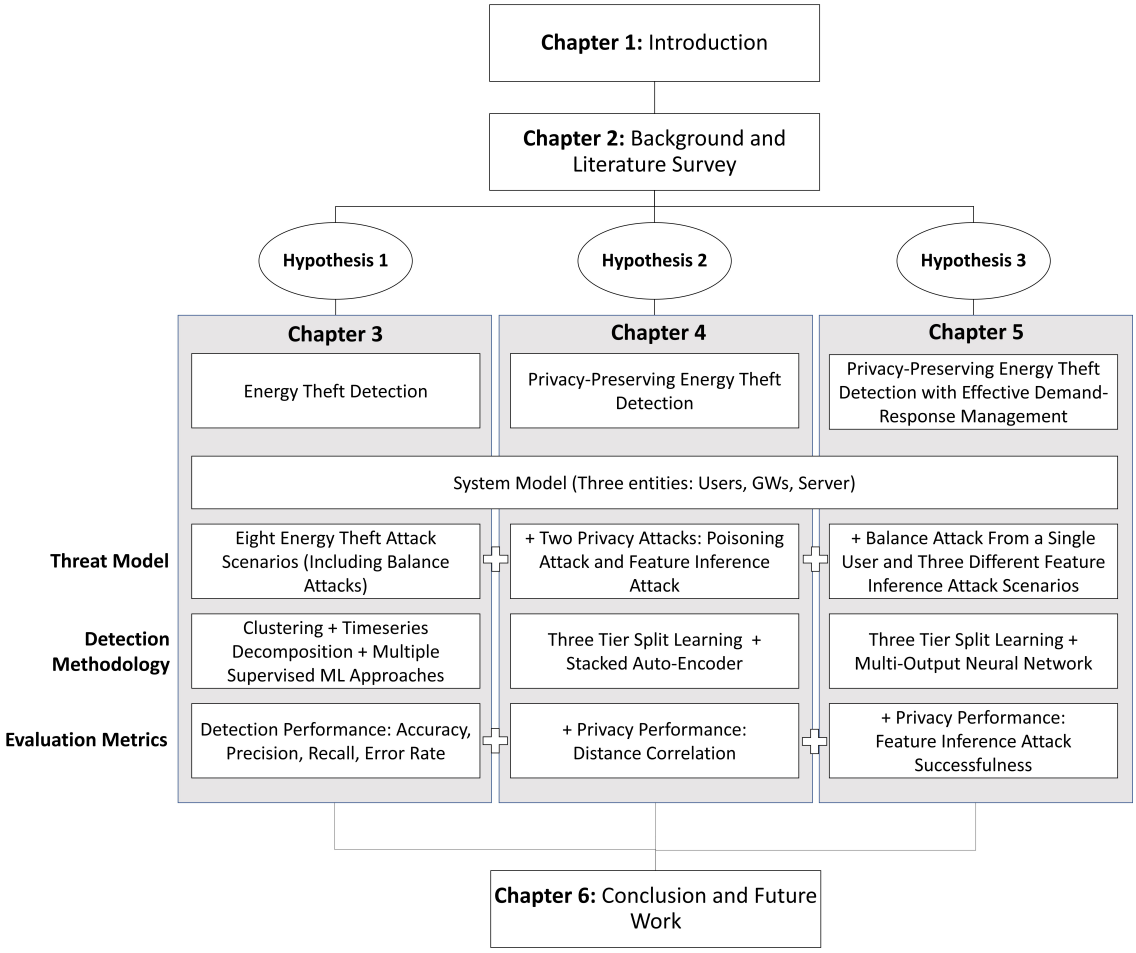


Figure 1.1: *Overview of the Thesis Structure*

In **Chapter 3, ML-Based Detection Model in the Presence of Prosumers**, we propose a cluster-based detection model that uses multi-source data features to detect energy thefts. We apply our proposed model to different energy theft scenarios, including those that prosumers can launch. The proposed model is empirically tested using several machine learning algorithms, and results can confirm *hypothesis 1*.

In **Chapter 4, Privacy-Aware Split Learning Based Energy Theft Detection**, *hypothesis 2* is explored through the implementation of a detection model for energy thefts that can preserve the privacy of users' data. This work introduces a new variant of a privacy-preserving ML approach, which we term Three-Tier Split Learning. This variant is needed to suit the smart grid's environment. Moreover, the model's security and privacy aspects are evaluated in different scenarios.

In **Chapter 5, Privacy-Enhanced Energy Theft Detection for Effective Demand Management**, we investigate *hypothesis 3* by integrating energy theft detection with demand management. We implement a multi-output system that can detect energy theft, estimate its magnitude and predict future energy demand. We also provide a thorough quantitative privacy analysis using two metrics: distance correlation and feature inference attacks.

In **Chapter 6, Conclusion and Future Work**, we summarise this thesis's major findings, restate the contributions achieved and highlight future work.

1.4 Associated Publications

The work reported in this thesis has appeared or submitted to appear in the following publications:

- Arwa Alromih, John A. Clark, and Prosanta Gope. "Electricity Theft Detection in the Presence of Prosumers Using a Cluster-based Multi-feature Detection Model." In 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pp. 339-345. IEEE, 2021. The URL for the electronic version of this publication is https://ieeexplore.ieee.org/document/9632322. This work is reported in Chapter 3.
- Arwa Alromih, John A. Clark, and Prosanta Gope. "Privacy-Aware Split Learning Based Energy Theft Detection for Smart Grids". In 24th International Conference on Information and Communications Security (ICICS 2022), pp. 281–300, Springer-Verlag, 2022. The URL for the electronic version of this publication is https://link.springer.com/chapter/10.1007/978-3-031-15777-6_16. This work is reported in Chapter 4
- Arwa Alromih, John A. Clark, and Prosanta Gope. "A Privacy-Preserving Energy Theft Detection Model for Effective Demand-Response Management in Smart Grids". This work has been submitted for publication and a preprint version can be found here https://arxiv.org/abs/2303.13204. Chapter 5 of this thesis reports on this work.

In addition to these publications, the generated dataset used in this thesis is openly available in our GitHub repository https://github.com/asr-vip/Electricity-Theft

Chapter 2

Background and Literature Survey

2.1 Introduction

The term "energy grid" typically refers to the Traditional Grid (TG) system, which is an infrastructure that supports four essential electricity operations: electricity generation, long-distance electricity transmission, energy distribution, and end-user power consumption. In the TG, energy stations dispatch electricity unidirectionally to distribution substations and finally to the end users. However, this outdated power infrastructure cannot meet the rising demand for services like demand response (DR), self-healing, real-time pricing, congestion management, dependability, and security. It is crucial to concentrate on the newest technologies to satisfy these demands and deliver safe, dependable, continuous electricity without power system blackouts. These features are all available in the future grid, commonly known as the Smart Grid (SG).

The smart grid is an advanced concept proposed at the beginning of the 21st century [4]. It is the evolutionary step towards reliable and efficient power delivery. SG networks are advanced technology-enabled electrical grid systems that incorporate information and communication technology (ICT) with smart meters, metering communication networks and meter data management systems to collect nearly real-time big energy usage data with a view to its subsequent analysis [4, 5]. ICT enabled two-way information and electrical flow between the grid's entities, facilitating the automatic distribution of electricity delivery. By utilising cutting-edge information and communication technology, the SG can produce, store and share energy whenever it is needed, just like we create and share information through the internet [6, 7]. It is important to note that the system had been made more

complex by integrating a number of technologies (as shown in Figure 2.1), including advanced metering infrastructure (AMI), energy supply systems, renewable energy sources (RESs), electric vehicles (EVs), and energy storage systems (ESS) all with the help of ICT [4, 8].

While SG has opened up new opportunities for better energy management, it has also created new potential problems. One of the significant difficulties is the threats to the system's security and data privacy. The most common security threats are those that result in considerable functional and monetary losses for energy utility firms on a global level, such as energy theft. Energy theft attacks raise serious issues for both providers and customers. Whether such attacks are carried out on a small or large scale, by a single user or multiple users, the losses will eventually affect everyone, including honest users. To maintain SG's activities, a security mechanism must exist to defend against them. Otherwise, customers may experience an electrical blackout that would disrupt daily activities, including the failure of heating systems, the absence of online payment systems, and many others. Moreover, the modernisation of the grid has significantly increased the amount of personal information exchanged in the system. This has created more opportunities for attackers to gain access to individuals' information and maliciously utilise it improperly.

Several surveys and reviews, such as [3, 9, 10, 11, 12, 13], have evaluated energy theft detection algorithms, the issues that occur during the detection functionality and the existing limitations of each detection category. However, these existing surveys focus only on the proposed detection mechanisms for energy theft and do not cover state-of-the-art energy theft attack techniques. They also consider energy theft as a sole problem and do not consider the implications that are faced either from the detection approach (such as violating customers' privacy) or those that arise from the theft act itself (such as compromising the demand management accuracy). Therefore, this chapter provides a more comprehensive overview of the different attack types, their detection strategies proposed in the literature, and a thorough evaluation of their strengths and weaknesses. This will facilitate further research and help in considering the existing limitations in this research area. We outline some background related to the smart grid, its architecture, security requirements and cybersecurity-related issues. We provide more details about energy theft attacks, outline research efforts that seek to defend against them and identify to what degree each research achieved its goals. We also consider evaluation methods, attack scenarios, and datasets used by other researchers.

The rest of this chapter is organised as follows: Section 2.2 focuses on background information about SGs' architecture and components. Sections 2.3 and 2.4 list the overall security requirements and security threats that are faced in the SG's environments. Section 2.5 provides a comprehensive description of energy thefts

and Section 2.6 highlights their implications over privacy. Section 2.7 discusses and categorises the algorithms used for energy theft detection systems. Next, the performance metrics and datasets used in the energy theft research area are described in Sections 2.8 and 2.9, respectively. Finally, Sections 2.10 and 2.11 present the existing research gaps, address the identified research questions and formulate the research's model.

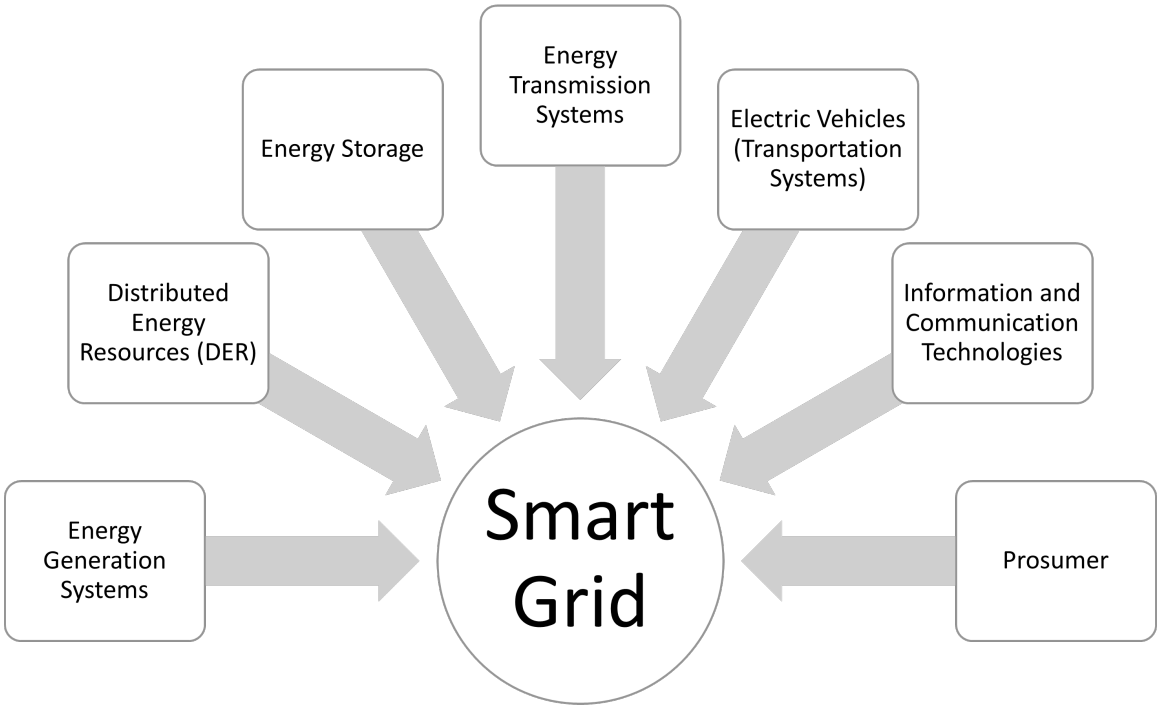


Figure 2.1: *An Overview of Smart Grid's Components*

2.2 SG Architecture and Components

According to the definition of smart grids, an SG is a cyber-physical system (CPS) that integrates legacy complex power and energy systems with information and communication technologies (ICTs). Being a CPS, the smart grid system can be treated as a three-component architecture, as shown in Figure 2.2 [14]. These components are:

1. A physical layer that comprises the physical infrastructure of electrical power systems.

2. A communication model that enables information exchanges between different elements of the system.

3. A control layer which includes all software managing and controlling the energy systems.

In the sections that follow, a detailed description of each component is given.

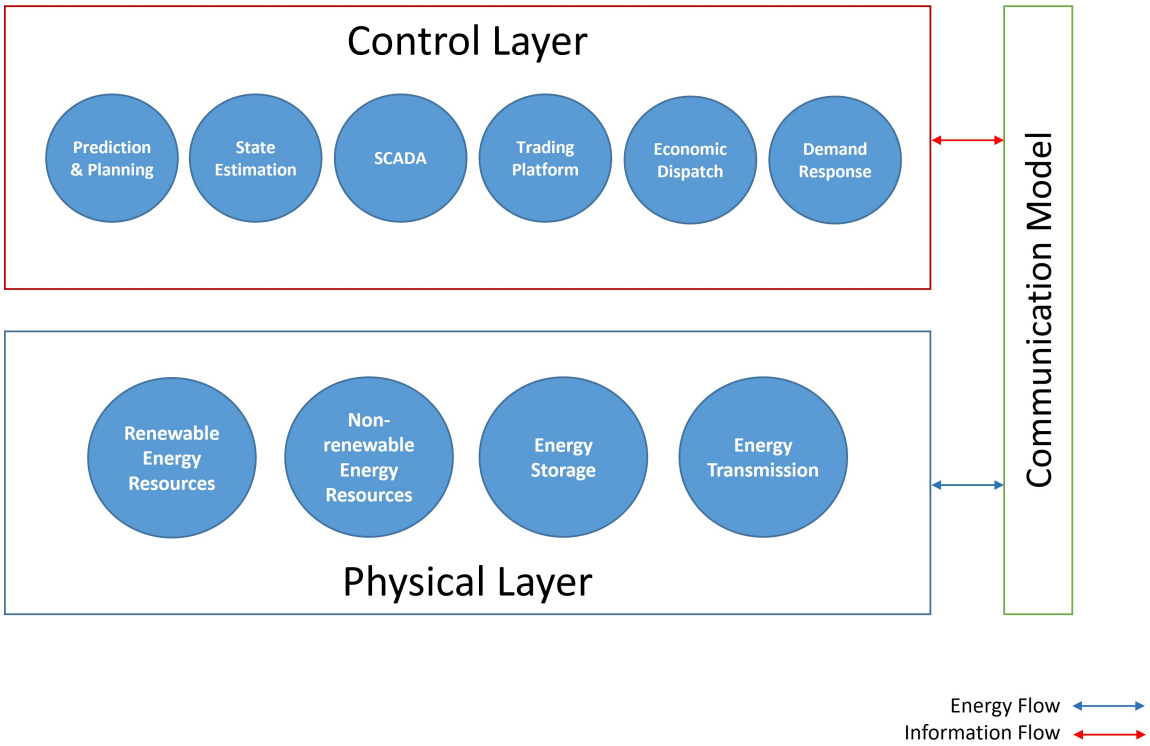


Figure 2.2: *An Architecture Reference Model for Smart Grids [15]*

2.2.1 Physical Layer

The smart grid's physical infrastructure comprises several power systems generally categorised into generation, transmission, and storage systems. The generation systems integrate renewable energy systems (RESs) along with traditional power plants to generate electricity to be delivered by the transmission systems to customers [16].

2.2.2 Communication Model

The communication network in a smart grid system can be divided into three subsystems, as shown in Figure 2.3. This network comprises three parts: the Home Area Network (HAN), the Neighbourhood Area Network (NAN), and the Wide Area Network (WAN). In the Home Area Network, a group of home appliances distributed energy resources (such as solar panels or small wind turbines) and energy storage systems are connected to a smart meter (SM) for basic data collection. Moreover, an in-home display provides the user with an interactive interface to control and manage all devices inside the HAN. The SM enables two-way communication to send and receive information from and to the utility. A NAN connects several HANs together with a gateway (also known as an aggregator or data concentrator) that resides at a substation. It is mainly located in the energy distribution domain. Therefore, it communicates over power-line communication (PLC). Finally, the WAN connects multiple gateways from different substations together to provide connectivity to the utility control system [5, 17].

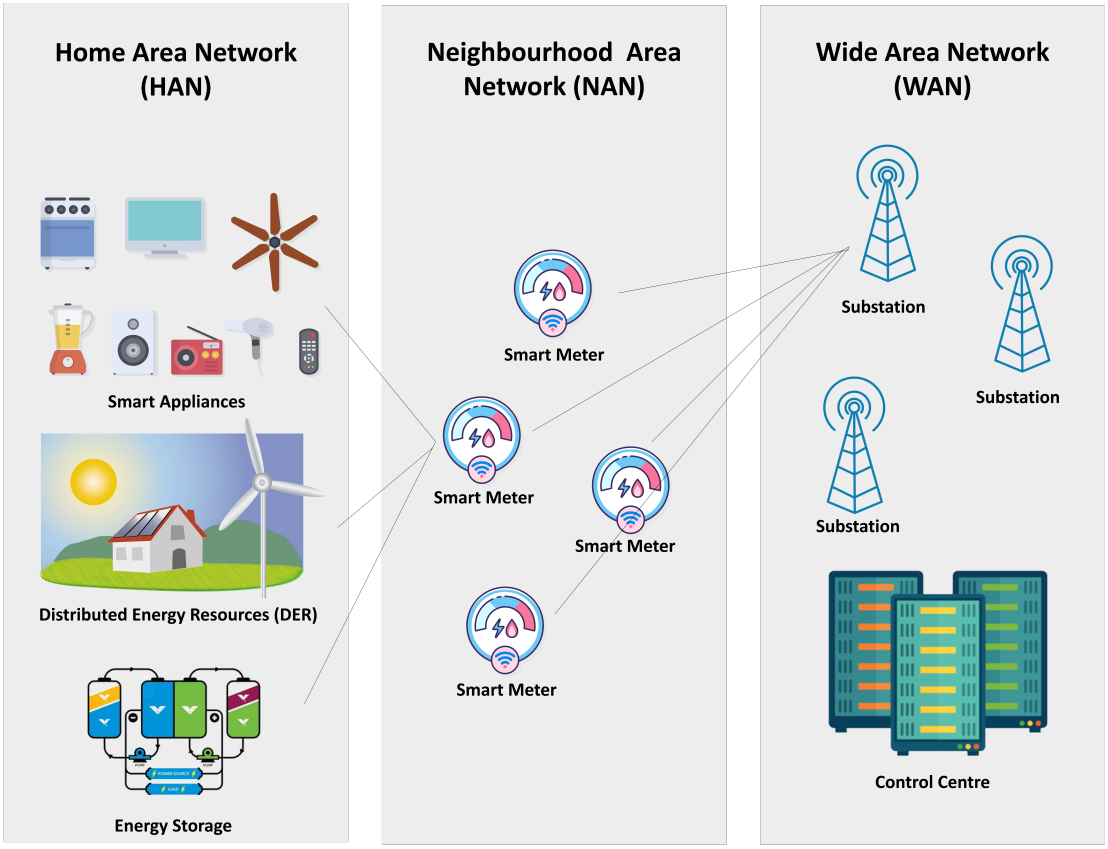


Figure 2.3: *Smart Grid's Data Communication Network [5, 17]*

2.2.3 Control Layer

This layer is responsible for controlling and monitoring the energy elements of the system. It has one major component, which is the Supervisory Control and Data Acquisition (SCADA) system. SCADA systems reside at control centres and are used to facilitate all decision-making processes in the smart grid infrastructure. SCADA employs various sensors and actuators that sense and send data to remote terminal units (RTU). These RTUs forward all information to a Master Terminal Unit (MTU) for further processing and analysis. Other elements exist in this layer, such as the trading platform, state estimation, prediction and planning, economic dispatch and demand response. All of these systems rely on data stored in the SCADA system to be analysed and communicated with different parties through the communication model.

2.3 Smart Grids Security Requirements

The smart grid's network is a critical and sensitive network that requires secure methodologies to deal with the cyber system and the communication infrastructure. Moreover, the communication model in SG systems handles the process of sending command information, consumption reports, prices and bids, billing and demand response controls [5, 15]. Hence, the sensitivity of these communications requires the SG to provide the following security goals:

- **Confidentiality:** Confidentiality is the security requirement that ensures that data are shielded from disclosure to unauthorised users and from eavesdropping. Confidentiality in itself may be of limited importance to the operation of an energy system, but it is closely associated with privacy, which is critical to customers due to the periodic energy usage data communication through the network. These data can reveal consumers' life patterns. The importance of privacy and its issues in energy systems will be discussed later in Section 2.6.
- **Integrity:** This requirement assures that the data and system commands are safe from unauthorised modification and alterations. This is particularly important since falsified and altered data or commands can enable an attacker to hijack the system and gain sensitive information, manipulate meters' readings, escalate privileges or access unauthorised system components.
- **Availability:** This ensures that all smart grids' data and systems are available at all times (or very close to that). The availability of energy systems is crucial due to the fact that generation and consumption need to be balanced.

- **Authentication:** An energy network consists of a large number of different entities whose identities need to be ensured to prevent user impersonation. Authentication is the method of ensuring the validity of a user's identity and the content of the information sent by the user.

- **Non-repudiation:** This property ensures that a component cannot deny the actions that it performed. This ensures that all false actions can be tracked to a certain individual; for example, an energy thief cannot deny responsibility for the actions he/she carried out.

- **Authorization:** This requirement ensures that permissions are given before any action is carried out. It is essential to share consumption data between components to determine demand and load management where only authorised components can read these data.

The importance of these requirements differs across domains; for example, the grid automation and control systems that ensure the industrial operation would favour availability and integrity over confidentiality since they are the main drivers for grid automation, industrial safety and environmental impacts. However, in the smart metering domain, confidentiality, or more precisely privacy, would be prioritised since consumption data is the most sensitive for individuals as it can reveal users' habits [5]. In our work, we give high importance to both integrity and privacy as they are the most impacted requirements by energy theft attacks.

2.4 Security Threats in Smart Grids

Since a smart grid is considered a complex CPS, complex security challenges have been revealed in the physical and cyber models of SG [16]. On the one hand, physical devices, smart meters and bulk power systems can be vulnerable to physical security threats. On the other hand, there are cyberspace vulnerabilities that affect the privacy and protection of the communication systems and the information at the software level [18]. According to the aforementioned factors, attacks on SGs can be categorised into physical and cyber, as shown in Table 2.1. These categories are discussed in more depth in the following subsections, with a brief look at the history of cyber attacks first.

Table 2.1: *Physical and Cyber Attacks Targeting Smart Grids*

Attack Type		Attack Name
Physical		1. Meter manipulation 2. Physical power lines cutting 3. Natural disasters
Cyber	Integrity	1. False data injection attack (FDIA) 2. Energy thefts
	Confidentiality	1. Traffic analysis 2. Release of message contents
	Availability	1. Channel jamming 2. Denial of service (DoS)

2.4.1 History of Cyber and Physical Attacks in Smart Grids

Over the past 40 years, several cyber-physical security attacks have been launched against the energy sector. These attacks had different levels of impact ranging from unnoticed information loss to losses of millions of dollars. According to the ninth annual report by Accenture/Ponemon [19], it has been revealed that cybercrimes in the energy industry have resulted in an average of 13.8 million dollars in losses. This places the energy sector in the top ten industries that suffer financial losses from cybersecurity attacks [19]. Cybercrimes against energy systems are not new; they began in the 1980s. A summary of publicly known attacks can be found in Table 2.2.

In 1982, the first major attack was announced when a massive Siberian gas pipeline explosion took place due to a trojan horse implemented in the control software from the United States [12]. The attack was categorised as a malicious update to firmware that influenced a single substation [20]. In 1994, a hacker managed to access the computers of the Salt River Project water facility in Arizona, U.S. and gain complete control of the SCADA system for five hours. During this attack, the attacker accessed/altered customer financial and personal records that cost the company around $40,000 [20, 21].

During the following ten years, several security incidents were recorded. Some were due to insider employee collaboration, such as the Gazprom incident in Russia.

Table 2.2: *Overview of Cyber Attacks in the Energy Sector*

Attack Name	Year	Location	Method of Operation	Impact
Siberian Pipeline Explosion [12, 20]	1982	Siberia, Russia	Trojan Horse	Physical damage
Salt River Project [20, 21]	1994	Arizona, U.S.	Root Compromise	Financial loss & data disclosure
Gazprom [21]	1999	Russia	Insider & Trojan Horse	Operation disrupt
Bellingham Gas Pipeline [21]	1999	Washington, U.S.	Misuse of resources	Human loss
Davis-Besse Nuclear Power Plan [21]	2003	Ohio, U.S.	Worm	Operation disrupt
Aurora Attack [12]	2007	Idaho, U.S.	False data injection attack	Financial loss
Stuxnet Attack [21]	2010	Iran	zero-day attack	Operation disrupt
Blackout [20]	2015	Kiev, Ukraine	False data injection attack	Operation disrupt affecting 22.5k customers
Aramco Malware [20]	2017	Saudi Arabia	Malware injection	Generation and delivery disrupt

Others were because of failures in the critical infrastructure system, as in the case of the Bellingham Gas Pipeline misuse incident in Washington, U.S. [21]. In 2007, the "Aurora" cyber attack was launched against a control system of a test generator. The attacker injected false commands to switch the circuits on and off, causing a desynchronization between the mechanical generator and the electrical grid. This desynchronization resulted in the explosion of the generator leading to a loss of one million dollars [12]. Following these attacks, the "Stuxnet" attack struck a nuclear facility in Natanz, Iran, in 2010. Stuxnet had exploited four zero-day vulnerabilities targeting the Microsoft Windows operating system. The attack changes a centrifuge's rotor speed, raising its speed and then lowering it, causing it to fail faster than normal [21]. Energy systems worldwide experienced more cyber-attacks during the next 10 years, i.e. between 2010 and 2020. Examples are the often-quoted attack on

a Ukrainian distribution grid operator in 2015 and the malware targeting industrial control systems of the Saudi Arabia energy infrastructure in 2017 [20].

Understanding the evolution of cybercrimes against energy systems over time can assist in the development of new techniques to mitigate their impact. The investigation of these cyber-physical attacks starts by first identifying their different types.

2.4.2 Physical Attacks

Physical attacks on energy systems refer to any actions taken to physically damage or destroy energy infrastructure or a specific area or location. These attacks are launched in an attempt to disrupt the energy supply. This can include acts such as bombing a power plant, cutting power lines or vandalising equipment [16]. An example of this form is the 2013 attack in California where a sniper targeted a Silicon Valley power substation [22]. These types of attacks can have serious consequences, such as power outages, equipment damage, and even loss of life. They can also disrupt the normal functioning of critical services such as hospitals, emergency services, and transportation systems.

The real challenge is that the infrastructure is geographically spread and distributed across the land. Thousands of miles of power lines, generators, and substations are in danger of physical attacks. Likewise, smart meters are installed in customers' homes and businesses. For example, a utility cannot prevent a motivated person from cutting down a transmission line or physically damaging a substation [16].

2.4.3 Cyber Attacks

Cyber-based attacks are attacks delivered through the system's control layer. They can be categorised based on the three basic security requirements, confidentiality, integrity and availability (CIA) [18], as follows:

- **Attacks against data confidentiality:** Confidentiality attacks are attacks that try to steal and have access to information that is meant to be secret within authorised parties. These attacks are also known as attacks against data privacy. Due to the sensitive nature of information in energy systems and the multi-hub routing nature of these systems, an adversary can eavesdrop on communicated reports and analyse traffic patterns. Compared with integrity

and availability attacks, confidentiality attacks could be seen as lower-risk attacks as they often do not directly impact the smart grid's functionalities. An attacker can sniff or wiretap a communication channel to get personal information, such as consumption data and bank information, without affecting the operation of the network. However, collecting this information about the system is the first step in any attack cycle. Moreover, data privacy is violated when information is sent in clear text. Privacy has seen more attention in recent years, especially with the huge amount of customer personal information revealed in recent leakage incidents [23, 24]. We note, however, that certain properties of confidentiality must be maintained, e.g. key management must uphold confidentiality since cryptographic keys often underpin the achievement of other security requirements.

- **Attacks against integrity:** Data reports and commands are one of the main factors in the control and decision-making process that is taken by the SCADA system at the control centre. Any false measurements or malicious commands can lead to catastrophic results. These false measurements are called false data injection attacks, and they are one of the major integrity attacks in smart grids. Several false data injection attack incidents have been reported in energy systems, such as the "Aurora" cyber attack and the Ukrainian energy distribution system attack, which were discussed previously in Section 2.4.1. Different types of data are communicated between components in the SG system. These data can all be vulnerable to manipulation and are listed below:

  1. Smart meter data.
  2. Power injection requests and bids.
  3. Price signals from the utility.
  4. Electrical data of the grid that represent real and reactive power flows, demand response capacity and voltage.
  5. Event messages data, such as outage alerts.

  When customers falsify smart meter readings in the system, this is referred to as an "energy theft attack", where the attacker's intention is likely to steal energy. Section 2.5 defines this type of attack in detail. In order to defend against integrity attacks, integrity needs to be provided using any integrity mechanism such as digital signing or hashing. However, these mechanisms alone are not sufficient as a compromised node could forge malicious reports along with the correct signature or hash. Another misconception about the defensive mechanisms against these attacks is that they can be solved using encryption. This is not entirely true, as confidentiality does not equate to data authenticity.

- **Attacks against availability:** Any disruption to the system's availability, such as in the case of denial-of-service (DoS) or distributed DoS (DDoS) attacks, may lead to significant economic losses (and further losses in critical domains). In these attacks, attackers typically send a large volume of packets to flood the network, which causes legitimate data packets to be lost. Jamming attacks also target the availability of the system. They aim to cause noise in wireless communication networks so that smart meters (edge devices) cannot connect to the energy infrastructure network. Such attacks result in packet loss. We also note the existence of more subtle forms of denial of service, e.g. low rate denial of service attacks, where a server's request buffer is maintained at a full level, causing service requests to be dropped, but where there is no "swamping" with requests. Attackers only need to keep a server 100 per cent busy to effect a DoS attack; they do not need to overload it many times. As mentioned in Section 2.3, availability is generally considered the most important cyber security requirement for power systems. Thus, effective defences should be made against these attacks. Traffic filtering, anomaly detection and channel hopping are some solutions [23].

2.5 Energy Theft Attacks

Energy theft can be defined as the illegal use of energy from electric providers without a valid contract or any act that leads individuals to not pay their electricity bills or pay less than they should due to meter reading manipulation. The quantity of electrical usage relies on the amount of power consumed for a certain duration of time. The amount of power consumed, i.e. real power, is the product of voltage, current, and power factors. Once at least one of these three factors is altered by dishonest clients, meters may be measured, recorded, or charged incorrectly. As stated before, energy thefts are a type of false data injection attack where an attacker manipulates the meter's measurements to make a change in the value reported. This manipulation of data can be done by a compromised sensor meter in the smart grid in various ways, such as: (1) a compromised customer's meter purposely forges its own sensed reading; (2) an en-route meter forges the report it is relaying to its parent; or (3) an aggregator meter modifies or drops the aggregated value it is passing to the base station.

Energy theft attacks can be launched for different durations of time. They can be launched as a one-off attack (interim) or for a continuous time. In an interim energy theft attack, the duration of the attack is a short time interval. This attack aims to inflict maximum damage in the shortest possible time. Such attacks can be detected by statistical anomaly-based detectors [25]. In the case of a continuous

attack, the attack is continuous, which means that once the attack starts, all the sensor readings are compromised from that point onwards. This attack is more interested in compromising and manipulating the system over a long period while avoiding detection. Such attacks are difficult to uncover with anomaly-based detectors alone. To defend against such continuous attacks, secure communication protocols could be a solution [25, 26].

Energy theft attacks can be one of the most critical and serious attacks launched against energy systems [27]. Therefore, the development of techniques that can protect and counter such attacks is essential to secure the operations of smart grid systems [28].

2.6 Energy Thefts and Privacy Issues

As defined previously, energy theft refers to any deviations between the actual electricity usage and the amount billed to a customer. The issue of privacy comes into play because energy companies may use various methods to detect energy theft, such as installing smart meters that can track usage in real time. The use of these technologies raises concerns about potential invasions of privacy. This is because these data can include private information about customers' energy consumption and billing information. These data can be vulnerable to breaches and misuse, and can be used without customers' knowledge or consent. To avoid these concerns, energy companies can implement various privacy-preserving techniques, such as data anonymization, data encryption, data aggregation, and access controls. These techniques can help protect sensitive personal information while still allowing it to be used for legitimate purposes, such as detecting energy theft. However, these techniques can also hinder the detection performance in the sense that the data is altered, thus making it harder to detect. This can be viewed as follows:

- Data Anonymization: Anonymizing personal data, such as removing names, addresses, and other identifying information, can make it more difficult to detect patterns of suspicious activity that are specific to individual customers. This can limit the ability to detect the theft's source.
- Differential Privacy: Differential privacy is a perturbation and randomization-based technique that relies on sanitising the data by adding noise before they are sent. The disadvantage of differential privacy is that it can only provide privacy to a certain level before it can lower the detection performance.

- Encryption-based Techniques: These techniques use encryption, such as homomorphic encryption or multiparty computation (MPC), to encrypt personal data. They could be utilised to analyse consumption patterns and detect energy theft from encrypted data. Unfortunately, those cryptographic-based techniques increase the computation and communication costs dramatically.
- Data Aggregation: Aggregating data from multiple customers can help protect individual privacy, but it can also make it difficult to detect the theft's source.
- Access Controls: Giving limited access to the data reduces the energy company's ability to detect thefts.

Therefore, finding an acceptable balance between energy theft detection and customer privacy protection is important. This balance is not an easy task, but it is essential in order to maintain trust and build a long-term relationship with customers. Moreover, some privacy techniques, such as differential privacy, homomorphic encryption and secure multiparty computation, can be combined and used to find the right balance between privacy and performance. Still, these can increase the complexity of the system.

In the next section, we will review the different categories of energy theft detection approaches proposed in the literature, including those that address privacy concerns.

2.7 Energy Theft Detection Techniques

In the literature, different strategies have been developed with the goal of detecting energy theft attacks. While these strategies differ, they can be divided into two main categories: non-machine learning-based methods and machine learning and deep learning-based methods [9, 29, 30]. It is worth noting that these methods are not mutually exclusive and can be combined to form a hybrid energy theft detection system. In this section, we will provide a detailed survey of these methods.

2.7.1 Non-Machine Learning Based Detection Methods

There are several non-machine learning energy theft detection schemes that can be used to detect energy theft. These schemes include game theory-based techniques, hardware-based techniques, and state estimation techniques.

Game Theory Techniques

Game theory is a widely used defence technique against cyber-physical attacks. In this technique, active attackers and reactive defenders are seen as the two players in the game [29]. Game theory provides powerful mathematical models and techniques for modelling and analysing the interactions between defenders and attackers. In the context of energy theft detection, energy thieves (the attackers) can use game theory to maximise the benefits of different types of energy theft without being caught, while energy providers (the defenders) use it to analyse the costs and benefits of different security measures over time in response to each theft.

Based on the game theory assumptions, Cárdenas et al. [31] proposed a Nash equilibrium-based game theory strategy to detect energy thefts. In this game, the goal of the attacker is to find the maximum amount of electricity to be stolen while minimising the expected likelihood of being detected. The goal of the utility is to maximise the probability of detecting thieves while lowering the operational cost of the detection algorithm. They also proposed a privacy-preserving demand response as a control theory problem that is solved with the goal of maximising the level of privacy by selecting the maximum sampling interval for smart meters. However, their proposed privacy-preserving control system cannot be combined with energy theft detection and is only applicable under many unrealistic constraints. Amin et al. [32] proposed another game-theory-based energy theft detection with the same previous two players in the system. The proposed scheme considers pricing and investment decisions by the utility, the amount of stolen energy, and the probability of being caught by the thief. Wei et al. [33] proposed a Stackelberg game theory-based model to identify energy thieves. A Stackelberg game is formulated between a single leader (the utility) and multiple followers (thieves) to characterise and analyse the interactions between them. The two actors in this game have opposite goals, i.e., the utility aims to maximise theft detection probability while limiting false positives. Whereas from the thieves' perspective, the strategy is to interact with one another in a non-cooperative manner to steal the optimal amounts of electricity without being detected. After formulating the game's equilibrium, a likelihood ratio test (LRT) is used to detect potentially fraudulent meters.

Game theory-based techniques are not very well-known in the energy theft detection research community. This is because they are based on the assumption that the number of players in a game is finite [9]. Another reason is that it is challenging to construct the utility's optimisation function as there is a number of trade-offs between all the required parameters that need to be taken into account when designing it [2, 34].

Hardware-Based Techniques

In hardware-based detection techniques, special types of physical equipment are installed at different places in the energy infrastructure to allow the identification of any theft activities [35]. These techniques range from simple physical security measures, such as locking each meter in a secure box, to more complex ones such as replacing traditional meters with special ones. These special devices are used to measure the current, voltage, magnitude and phase angle at fixed intervals from multiple locations to be analysed for any inconsistencies [2]. In Grewal et al. [36], the authors proposed a metering-based theft detection that works by deploying enhanced prepaid energy meters in customer premises. This hardware-based theft detection system monitors the power consumption with respect to the load where two current transformers are connected before and after the energy meter for theft detection. If any change exists between the two current readings, an alarm is sent, indicating possible power theft. A prototype of this proposed scheme was developed to test its applicability and efficiency, and preliminary results showed that the detection rate (alarm rate) was almost 90%. The main drawback of this work is that it was tested in a small circuit; therefore, it is unclear whether it can scale to large systems. A similar approach was also proposed by the authors in [37], [38], and in [39], where two sensors are placed to measure the amount of current at both ends of the energy meter. When a difference between the two values occurs, energy theft is identified.

Hardware-based detection methods are simple and have the ability to detect any illegal behaviour in consumption. However, the cost of deploying extra pieces of equipment around the whole network is expensive. Therefore, it is necessary to strategically choose the right number of these physical devices along with their appropriate deployment place [2].

State Estimation Techniques

State estimation is a technique that uses mathematical algorithms, such as Kalman filter, to estimate the current state of the system at various points. Energy thefts are detected by comparing the estimated state variables with the actual measurements. Therefore, estimating system states accurately is crucial for the correct decision-making in energy networks. These estimates rely purely on measurements taken from various sensors in the transmission lines, which include: active/reactive power injections (P/Q), branch power flows (S), and voltage angle/magnitudes (θ/V). The relationship between these measurements and the state variables to be estimated is expressed as follows:

$$z = h(x) + e \tag{2.1}$$

where z is the measurement vector (known), including active and reactive power flows (P and Q), and x is the system state vector (unknown quantities) for which the equation must be solved, and it includes the voltage magnitudes and phase angles (V and θ). e denotes the noise vector, which has a Gaussian distribution, and $h(x)$ denotes the mapping matrix between measurements and state variables. The precise form of $h(x)$ is determined by the grid structure and line parameters. To estimate the new state of the system, equation 2.1 can be solved using a weighted least squares (WLS) method. In this method, the vector of estimated state variables x is obtained by solving the following optimisation problem:

$$min \quad J(x) = \frac{1}{2}(z - h(x))^T \ W(z - h(x)) \tag{2.2}$$

where W is a diagonal matrix represented as $W = diag(\sigma_i^2, 0)$ and σ_i^2 is the variance of the measurement errors associated with the i-th meter. Estimating the system state requires a large number of measurements which can be susceptible to errors and faults. Therefore, bad data detection (BDD) is one of the essential functions in state estimation that is implemented to detect and eliminate these bad data [2, 34].

When an energy theft attack is crafted against state estimation, it is important to manipulate the right state measurements so that the bad data detection module is not triggered. This means that an energy theft attacker should manipulate the measurements and state estimation data of several buses and lines in a coordinated manner [40]. One important aspect of constructing a valid energy theft attack without being detected is if the attacker has sufficient knowledge of the target system. By knowing the system configuration and state parameters, the attacker can craft an undetectable false measurement that is injected into the system [2, 34].

A great number of state estimation techniques and BDDs have been proposed to detect energy thefts. Huang et al. [41] have proposed a detection technique for detecting electricity thefts using state estimation. It consists of two phases: the first one is to estimate the system state measurements using the WLS method. After that, the normalised residuals (difference between estimates and actual meter measurements) are used to localise the area where anomalous usage occurred. In the second phase, an analysis of variance (ANOVA) is used along with the customers' historically validated usage to detect suspected energy thieves. ANOVA was also used as the last step to identify energy thieves by the authors in [42]. However, in their proposed scheme, the authors used semi-definite programming to get the state estimation solution. This helps in finding the global optimal solution for the system's state rather than the local one (which is obtained from the WLS). After estimating the state of the system, the residuals are considered and combined with a historical analysis of a customer to detect electricity thefts. The two proposed techniques could successfully identify energy thefts; however, the proposed scheme

in [41] can only detect an individual malicious meter at a time and the work in [42] was only tested to detect up to two malicious meters. In [43], the authors proposed a model-based technique to detect and localise data theft attacks in microgrids. They used a stochastic Petri Net (SPN) to model the system's operation with three modes and transitions. Any disturbance in the electrical resistance will trigger an alert where suspected smart meter readings are forwarded to a Meter Data Management System (MDMS) for detection and localization. In the MDMS, Singular Value Decomposition is used to detect and localise data theft with an accuracy of 98%. This technique minimises transferring data to the MDMS in order to protect customer data privacy. However, as with most state-estimation techniques, the implementation and maintenance costs are high.

With regard to privacy-preservation, Salinas et al. [44] introduced a privacy-preserving state estimation-based detection system in 2013. In fact, their work is considered the first to study the issue of privacy in energy theft detection. The authors designed three distributed privacy-preserving approaches to identify fraudulent users based on two well-known decomposition algorithms: LU and QR factorization. These algorithms, just like WLS, can solve a linear system of equations corresponding to the consumers' energy consumption data (i.e., a data matrix) that must agree with the total load consumption measured by the collector at each time interval. Although this was the first work to look at privacy in energy theft detection, the work did not consider the issue of technical losses. Following their work, Salinas and Li [45] have also proposed another privacy-preserving energy theft detection based on state estimation. In their proposed work, the authors introduce a decomposed, loosely coupled version of the Kalman filter that can hide energy measurements and preserve users' privacy. However, this proposed loosely coupled filter can only be employed in small-scale microgrids since the complexity would increase as the size of the grid increases. In addition, according to [46], the proposed scheme can only detect continuous thefts with consecutive reduction reads (i.e. when the meter readings show a consistent decrease in consumption over a period of time), while as we saw in Section 2.5, energy thefts can be of different types including interim ones. The energy theft detection scheme proposed in [47] is yet another state estimation-based detection system that preserves privacy. The authors use a recursive filter based on state estimation to estimate energy consumption for all users and compare it with the true reading. If the difference is larger than a predefined threshold, then the reading is flagged as abnormal. In their work, the authors used the Number Theory Research Unit (NTRU) algorithm to encrypt users' data and preserve users' privacy. The simulation results show that their algorithm achieves an accuracy of more than 92%. However, the scheme introduces communication and computation overhead. Another weakness of this proposed scheme is the assumption that aggregators are trusted entities which is not always the case. Most aggregators are third-party companies

that are not governed by any authority.

State estimation techniques are the second most widely used method for energy theft detection [34], after ML-based approaches. This is because they achieve a higher detection rate compared to other detection approaches without the need for large historical data [2, 34]. However, these methods suffer from the following limitations [2, 34]:

- State estimation-based techniques require the detailed topology and parameters of the energy network, which is usually hard to get and update regularly.
- Most techniques can only localise the region of the energy theft and cannot pinpoint a particular customer.
- Most of these techniques were only evaluated in terms of detection rate and some were not evaluated at all. This can hinder the ability to compare their performance with other proposed energy theft detection approaches.

2.7.2 Machine Learning Based Detection Methods

With the rapid development of machine learning (ML) algorithms, several contributions have employed them as effective ways to detect anomalies. By using machine learning techniques, patterns of normal electricity usage are generated and the real-time operation and data of the system are monitored in order to detect any anomalies. These algorithms are increasingly being used because of their ability to be scaled to large systems and their low computational costs [48]. Machine learning approaches such as supervised learning, semi-supervised learning, unsupervised learning, especially deep learning and reinforcement learning have all been brought to bear.

We start by reviewing some work that used supervised machine-learning techniques to detect anomalous data in electricity usage. Gunturi and Sarkar [49] proposed to use a supervised machine learning algorithm to detect non-technical losses based on ensemble ML techniques. Ensemble ML models combine multiple ML approaches into one predictive model to boost the detection rate and lower the error rate. In their study, the authors found that a bagging-type ensemble ML approach, which takes the average result of several independent MLs, performs better than a boosting one. A special type of boosting ensemble ML called extreme gradient boosting (XGBoost) was used in Buzau et al. [50] to detect energy thefts. XGBoost is a scalable implementation of a decision tree boosting system that works

by combining multiple decision trees to create a more powerful model. In this study, the authors used two types of data as inputs to their ML model: smart meter data along with contextual (auxiliary) data. Results showed that the XGBoost model is robust when the dataset is imbalanced. Another recent study [51] also used XGBoost. The study tested the proposed model with both balanced and imbalanced datasets and the results showed that the proposed method achieves good performance in both scenarios. The study, however, had multiple assumptions, such as the availability of honest readings for a long period of time, that restrict its real-world application. Other gradient-boosting classifiers were used and compared in [52]. The authors used three different classifiers: XGBoost, categorical boosting (CatBoost) and light gradient boosting method (LightGBM). In this work, the authors focused on the feature engineering part to improve detection performance as well as time complexity.

Authors in [53] proposed an electricity theft detection that is based on artificial neural networks. They proposed a wide and deep convolutional neural network (CNN) that consists of two major components: the wide component uses one-dimensional (1D) consumption data to calculate the output, and the deep CNN component transforms the 1D consumption data into two-dimensional data based on the 7 days of the week. The deep component has several connected layers that analyse data. This technique was evaluated against conventional ML techniques where it showed better AUC results. However, the paper did not investigate the appropriate choice of the number of neurons, number of filters and number of epochs. Another work also employed CNN in their electricity theft detection. In [54], the authors used it to automate feature extraction, combined with a long-short-term memory (LSTM) model to detect energy thieves. Their work achieved a plausible accuracy rate of 89% but a lower detection rate (recall) of around 87%.

Supervised machine learning techniques are the easiest and most accurate of the three categories of machine learning. However, they require the use of labelled datasets which is usually hard to acquire in anomaly detection domains. Hence, the use of semi-supervised ML methods represents a more practical setting for energy theft detection. Taking this into account, Hu et al. [55] suggested the use of a semi-supervised technique to address the issue of depending on a huge number of labelled data to train a classifier. The proposed work uses both labelled and unlabelled samples to train a feature extraction network (FEN) model to handle high-dimensional data and extract features, and a denoising auto-encoder (DAE) to detect energy thefts.

As discussed before, the lack of labelled data and the imbalanced distribution between anomalous and real samples in energy theft datasets have introduced the need to use unsupervised ML instead of semi-supervised or supervised learning. Unsupervised learning algorithms do not rely on labelled data and can be used to

discover patterns to identify anomalies in an adaptive and flexible manner, even in the presence of missing or corrupted values. Several types of unsupervised learning algorithms exist, such as clustering techniques, one-class classifiers, dimensionality reduction techniques and auto-encoders. Zanetti et al. [56] proposed a detection system that uses unsupervised clustering algorithms to construct short-lived consumption patterns. These short-lived patterns represent the consumer's profile for a short period. They are then used to detect any anomalies in the current consumption rates. The advantage of using short-term periods instead of long-term ones is that natural consumption rates change quickly and collecting short-term data reduces the vulnerability of violating data privacy. The detection system starts by tuning itself to the most suitable pattern duration (ranging from 1 day to 2 weeks). Then it starts the validation process using three unsupervised learning algorithms: fuzzy C-means (FCM), K-means and self-organising map (SOM). The results show that there is a trade-off between maximising the theft detection rate and minimising the false positive rate which costs more to handle than the theft itself. Therefore, maximising the F-measure is a better approach if we would like to improve the utility profit. Another clustering-based algorithm was used by Zheng et al. [57]. In their work, Zheng et al. used a density-based clustering algorithm with a distance matrix to identify unusual consumer profiles. The authors used a synthetic dataset in which abnormal load profiles for six malicious types of energy thefts. They tested their work using the following evaluation metrics: Area under ROC Curve (AUC), accuracy and F1-score, and results showed that the proposed density-based clustering technique outperformed other well-known clustering models in detecting electricity theft. A dimensionality reduction unsupervised ML technique, called principal component analysis (PCA), was used in [58]. The PCA-based detection technique was used to identify three attack scenarios of electricity theft by extracting critical features that can explain variations in the data monitored. After that, an anomaly score threshold is calculated using historical data. The results of this method indicate an average detection accuracy of 89.2% of all different attack scenarios. The work can be improved by optimising the choice of threshold values.

Privacy Preserving Machine Learning

Typical ML methods use vast amounts of data without any consideration for data privacy. However, there has been a recent introduction to a special type of ML technique, called Privacy-Preserving Machine Learning (PPML), that aims to protect privacy. The main idea of PPML is to allow ML models to be trained without the need to disclose private data in its clear form [59]. The PPML approaches fall largely into two sub-categories: cryptographic-based ML approaches and distributed-based ML [60]. In cryptographic-based ML approaches, traditional privacy-preserving

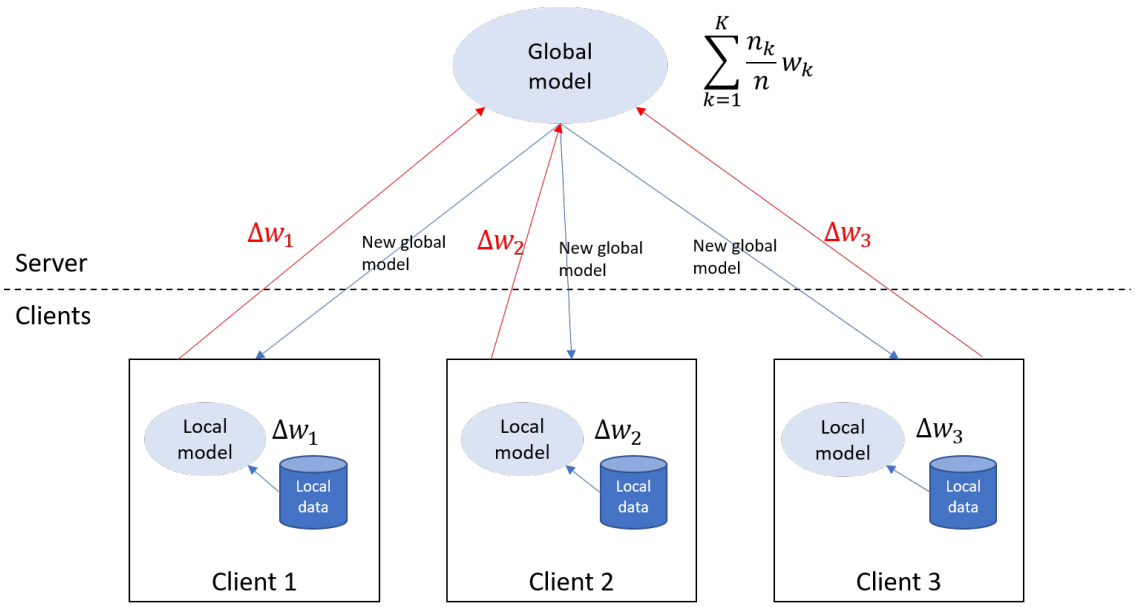


Figure 2.4: *Federated Learning*

techniques, such as differential privacy methods and encryption techniques that were introduced in Section 2.6, are added to typical machine learning algorithms in order to make them privacy-friendly [59]. However, they either provide privacy to a certain level or increase the computation and communication costs dramatically. An alternative to these techniques is the use of decentralised or distributed-based ML algorithms where training is done collaboratively between the system's entities [60]. Two major methods were introduced in this category: federated learning [61] and split learning [62].

Federated learning (FL) is a distributed machine learning algorithm that was introduced in 2016 by Google researchers [61]. The idea of federated learning is to build a global model based on clients' local models without the need to access their raw data. As illustrated in Figure 2.4, federated learning starts by sending an initial model to the clients where each client updates it based on its private data. After that, the weights of the update are sent to the server, where they are aggregated together to form an updated set of weights. Clients then download the updated weights and this process repeats until the model reaches convergence.

Another framework for distributed learning is split learning (SL), also known as split neural networks. This framework was developed by MIT to offer decentralised training for a model without sharing raw data by the clients [63]. In the basic form of split learning, a neural network model W is split into two parts W_c and W_S as shown in Figure 2.5. This aims to provide privacy protection for the client whilst

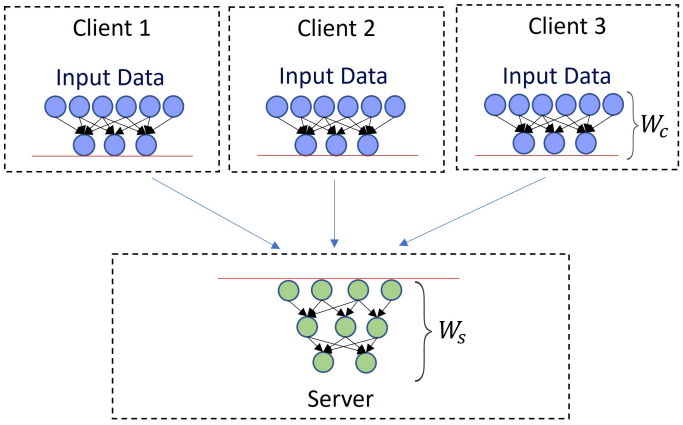


Figure 2.5: *Split Learning Setup Showing the Distribution of Layers Across Clients and a Server*

minimising the computational load. The first part of the network, W_c, resides on the client system and the remaining part W_S resides on the server side. These parts are called the client-side network and the server-side network respectively. Both the clients and the server train their part of the model separately where the process starts at $t = 0$ with the client data as the input layer, and then proceeds until the split layer is reached. The output of the split layer at client k, called activations $A_{k,t}$, is forwarded to the server to continue the training process. The server completes a full round of forward propagation to obtain the set of activations of the last layer $A_{S,t}$. The server now starts a backpropagation round from the last layer up to the cut layer where the gradients at the cut layer $\nabla\ell(A_{S,t}; W_{S,t})$ are sent back to clients. At the client side, the remainder of backpropagation is completed where W_c weights are updated for $t + 1$. This process is continued without the need for the parties to exchange raw data until the distributed split learning network converges. The complete algorithm of split learning can be found in Appendix B. Split learning is fairly new, and has not been applied in the context of smart grid security.

In the context of PPML, several works have been proposed for energy theft detection. In [64], the Paillier crypto-system was used to preserve the privacy of their proposed energy theft detector. Euclidean distances between energy readings over a day were used to detect abnormalities and frauds without revealing any valuable information. Another Paillier-based privacy system was introduced by Yao et al. [65]. In their security and privacy analysis, the authors state that the proposed detection algorithm achieves confidentiality, integrity, and data privacy by using encryption and digital signing. However, it is known that this is entirely dependent on the encryption mechanism strength. Nabil et al. [66] proposed a secure multiparty computation-based energy theft detection to preserve the privacy of energy readings.

The scheme uses secret-sharing techniques to allow smart meters to send masked data. Moreover, the use of secret sharing allowed the aggregation of data before sending them to the system operator. The detection of energy thefts is done online, where the smart meter and the system operator need to run a CNN model. Although the results suggested an accuracy of over 90% using different CNN architectures, the use of cryptographic techniques to preserve privacy introduces high communication and computation overheads. To overcome the need for running the detection model in both parties in parallel, the authors in [67] used a functional encryption (FE) algorithm to encrypt energy readings where energy theft detection is done without revealing the individuals' readings. Functional encryption is a relatively efficient cryptosystem that allows performing computations on encrypted data without the need to decrypt it. Although FE is assumed to be efficient in terms of communication and computation, it requires an extra step where a key distribution centre needs to generate and distribute keys for all participants in the system.

Wen et al. [68] have designed a federated learning-based energy theft detector with multiple local detection stations trained in a federated fashion. The model is then used to detect energy thefts from local users. To preserve the privacy of the local users' data, a local differential privacy algorithm is used to distribute the energy usage data of the grid's users. While this federated approach can preserve privacy, it introduces additional communication and computation complexity. Additionally, the scheme requires installing additional detection stations in the system. Another federated learning solution was introduced recently in [69], where a novel federated voting classifier, namely ensemble learning, is used. This scheme assumes that the use of federated learning preserves privacy. However, it has been proven that FL on its own cannot guarantee high levels of privacy and is very vulnerable to poisoning attacks, feature leakage or reconstruction, model extractions, and label inference attacks [70, 71, 72]. Recently, a blockchain-based privacy-preserving energy theft detection was proposed in [73]. Energy thefts are detected by comparing the aggregated consumption reports with the energy supplied. Users share their energy consumption privately using energy contracts in a ledger.

2.7.3 Hybrid Solutions

A hybrid-based energy theft detector combines techniques and algorithms that fall under two different categories. The use of multiple methods and techniques improves the accuracy and reliability of energy theft detection, however, it can increase the cost and complexity of the system [35].
An integrated system for detecting energy theft attacks was proposed by Messinis et al. [74]. This scheme uses two techniques: (1) a supervised ML method, SVM,

is used to detect energy thefts with minimal data; and (2) a state estimation-based technique that is based on voltage sensitivity analysis is further used in an attempt to estimate the time and extent of the thefts. The next study, [75], combined state estimation with machine learning in a two-step energy theft detection scheme. The first step is to detect energy thefts using a static state estimation technique that uses root squared percentage error as the residuals to be compared with the actual measurements. Whenever the percentage of error is above 10%, that specific region is further analysed to detect energy theft consumers. The next step uses the consumers' data from the suspected region to form Self-Organising Maps (SOM) that are used as inputs to a neural network-based detection model. The main weakness of the study is the inability to identify and localise the origin of the attack.

2.8 Evaluation Metrics

Evaluating the performance of an energy theft detector is a crucial step when proposing it. Different detection methods are evaluated using different measures, however, the most dominant evaluation metrics are those that evaluate the proposed model as a type of classifier. The performance of classifiers or anomaly-based detectors is usually calculated using a confusion matrix. Table 2.3 shows the definition of confusion matrix where True Positive (TP) is the number of intrusions that are correctly identified as anomalies. The False Positive (FP) denotes the number of normal records that are incorrectly identified as intrusions. The True Negative (TN) is the number of normal records that are correctly identified as normal and finally, the False Negative (FN) denotes the number of intrusions that are incorrectly identified as normal.

Table 2.3: *Confusion Matrix*

		Predicted	
		Normal	Attack
Actual	Normal	TN	FP
	Attack	FN	TP

Derived from the confusion matrix, there is a great number of well-known evaluation metrics that are used to evaluate the detection model. Table 2.4 presents a list of these widely used metrics along with their notations. Out of all these evaluation metrics, accuracy is the most known criterion. It calculates how many samples were classified correctly out of the total sample population. It is the simplest metric to evaluate and interpret because it is a single number that summarises the model's

capability. However, it is not a good performance measure for imbalanced datasets where class distributions are severely skewed. This is because it does not distinguish which class was correctly classified and it gives equal importance to false positives and false negatives [35]. In such situations, recall, precision, false positive rate (FPR), and F1-score are more popular metrics to use as they are more informative.

The recall, also known as detection rate (DR), sensitivity, or true positive rate (TPR), calculates the ratio between the number of correctly detected attacks to the total number of attacks. It is one of the most widely used metrics to evaluate and compare energy theft detectors. Precision and FPR are the other two reported metrics for energy theft detection. Precision reports the number of correctly detected attacks divided by the number of total detections. FPR, also known as false acceptance rate (FAR), is another important metric which calculates the number of falsely classified attacks over the total number of normal records. The F-score or F1-score is an evaluation metric that is used to evaluate systems that have binary classification. F1-score calculates the balance between precision and recall and thus can be considered as the harmonic mean of the two. It is mostly useful in cases of imbalanced data sets. Other metrics, such as error rate, the area under the curve (AUC), and mean average precision (MAP), have also been used to evaluate the performance of detectors in energy theft research. AUC, in particular, is a commonly used metric that provides the overall performance of a detector using a single value measuring the area under the receiver operating characteristic (ROC) curve, which plots the TPR against the FPR at various threshold settings [2].

A good detection model should have a high detection rate (DR) and a low FPR. This is because it is usually expensive to deal with false detections as they require technicians to have onsite inspections.

The above were the metrics used to evaluate the performance of energy theft detection. In addition, it is important to evaluate the privacy of the energy theft detection schemes, especially as privacy preservation is a challenging issue that was addressed in many proposed schemes. In the privacy-preserving energy theft detection domain, most of the literature analyses privacy theoretically and does not measure it by quantitative measures. This is because it is a complex and multifaceted concept that is evaluated differently in different domains. Privacy is usually measured by the properties or the parameters of the privacy-preserving technology used (e.g., the k in k-anonymity and the ϵ in differential privacy)[76]. On the other hand, the success of privacy attacks, such as feature inference, model extraction, and label inference attacks, is usually used as a privacy measure when PPML approaches are applied [77, 78]. However, these privacy parameters and privacy attacks cannot always be used as measures for every privacy-preserving technique. For example, model extraction and label inference attacks are difficult to launch and impractical in

Table 2.4: *List of Performance Metrics Used to Evaluate Energy Theft Detectors*

Metric Name	Definition	Representative References
Accuracy	$\frac{(TP+TN)}{(TP+TN+FP+FN)}$	[43, 45, 47, 54, 57, 58, 65, 66, 67, 68, 69, 73, 74]
Recall - DR - Sensitivity - TPR	$\frac{TP}{(TP+FN)}$	[36, 49, 51, 52, 54, 55, 56, 58, 66, 67, 69, 74, 75]
Precision	$\frac{TP}{(TP+FP)}$	[49, 51, 54, 69]
FPR - FAR	$\frac{FP}{(FP+TN)}$	[51, 52, 55, 56, 58, 66, 67, 74, 75]
F1-score	$2 \times \frac{(Precision \times Recall)}{(Precision+Recall)}$	[49, 54, 56, 57, 64, 69]
Error Rate	$\frac{(FP+FN)}{(TP+TN+FP+TN)}$	[36]
AUC	The area under the ROC curve	[33, 49, 50, 51, 53, 57, 66, 67, 68, 74]
MAP	The mean average precision for each class	[53]

a split learning architecture [79]. Therefore, they cannot be used as privacy measures. The issue of finding a unified privacy measure is highlighted in Section 2.10 as one of the limitations and open problems of the current energy theft detection literature. It is important to assess and measure privacy using unified numerical and statistical methods as it helps to objectively evaluate the privacy level provided by different privacy technologies and to identify areas for improvement [76].

It is worth mentioning that detection performance and privacy preservation degree are antagonistic metrics where the improvement in one of the two metrics will result in the reduction of the other. Thus it is important for the implementer (the energy utility) to choose the perfect balance.

2.9 Datasets and Energy Simulators

Due to privacy concerns, there are only a few public datasets that provide energy consumption and real energy theft incidents. Electric companies are unable to publicly provide detailed information and energy consumption statistics of energy

thieves, making it difficult for academics to collect genuine data to study. Many of the public datasets include only honest (real) consumption samples and do not have any malicious ones. Therefore, several existing works of literature have followed the same design approach of Jokar et al. [46] where data theft scenarios are synthetically added to a dataset in order to use them for training and evaluating their detection model. Table 2.5 summarises the most well-known energy consumption datasets, including the number of customers, reporting frequency and type of data included.

The first and most widely used dataset is the one released by the Irish Commission for Energy Regulation (CER) Smart Metering load profiles [80], which contains the consumption data of over 5000 residential and enterprise users. The consumption is reported at half-hourly intervals during 2009 and 2010. A downside of this dataset is that it contains only honest profiles and reports only the consumed real power at a half-hourly rate. The second widely used dataset is the State Grid Corporation of China (SGCC) (the largest electricity utility in China) [53]. This dataset is the first to include realistic labelled data, where each user is labelled as honest or a thief. The dataset contains the consumption data of 42,372 users from 1 January 2014 to 31 October 2016. However, the consumption is reported only once a day, making it difficult to identify the exact time of theft [81].

There are other datasets that are not as popular as the previous two. One is the UMass Smart* Project [82], which reports the electricity usage data for 443 anonymous homes located in a microgrid in Western Massachusetts. The data were collected every minute for the period of one day. However, the data from three homes were collected for the duration of a whole year in 2012. The last dataset is the Low Carbon London Smart Meter Trials Dataset [83], which contains half-hourly consumption data of 5567 houses. This data was collected between November 2011 and February 2014.

The drawback of these real datasets is that most of them lack the contextual data that might affect the consumption of a user, such as the floor area of the residency, location and weather conditions. Moreover, one of the drawbacks that led us to create our own dataset is the lack of a dataset that includes both prosumers and consumers.

As stated, previous studies that aimed to detect energy theft have mainly focused on analysing consumption reports only and finding periodical patterns for every customer. However, with the introduction of prosumers as a new actor into the electrical system, we require more data from other sources to be analysed in order to detect abnormalities. For example, there is a clear correlation between each prosumer's energy output and the DER's geographical location. Other correlations between the time of the day, the type of DER, its size and the amount of generated energy should be taken into consideration. This is why we need to have data from

Table 2.5: *Summary of the Reviewed Public Datasets*

Dataset	Number of Customers	Report Frequency	Includes Real Thefts	Includes Prosumers	Representative References
Irish Smart Energy Trial Dataset	5000+ residential and enterprise customers	Half-hourly	✗	✗	[49, 51, 52, 55, 56, 57, 58, 66, 67, 74, 75]
SGCC Dataset	42372 customers	Daily	✓	✗	[47, 53, 54, 65, 68, 69]
UMass Smart* Project Dataset	443 customers	Minute-level	✗	✗	-
Low Carbon London Smart Meter Trials Dataset	5567 customers	Half-hourly	✗	✗	-

multiple sources. The following is the possible set of different sources for data that can be used to aid the detection of energy theft attacks:

- Consumption Data: Smart meters monitor and report a number of different electrical parameters. Electrical parameters are classified into basic parameters and derived parameters. The basic parameters are voltage (V), current (I), and frequency (Hz), while derived parameters are active/real power (P), reactive power (Q), apparent power (S), displacement power factor (dPF), apparent power factor (aPF), active/real energy (Pt), reactive energy (Qt) and apparent energy (St). Table 2.6 lists all of these parameters and their equations. Previous research that is done to identify energy theft only considered the consumed real energy. However, having access to different power parameters at no extra cost opens the opportunity for us to use different measurement types and their relationships to increase the possibilities of theft identification as Laughman et al. [84] argues that merging different type measurements to analyse power can give higher accuracy in regards to event detection in general.
- Generation Data: As with consumption data, different types of electrical parameters are reported to the system. All of these parameters along with their relationships are to be monitored as a time series in order to identify if abnormal high values are present.
- Geographical Data: these include the address of the customer and his/her GPS location coordinates

Table 2.6: *Smart Meter Multidimensional Data and Their Description*

Parameter Name	Description	Unit	Equation
Voltage (V)	The difference between two points in a circuit. In most countries, it is equal to 220 volts.	V	N/A
Frequency (f)	The number of cycles per second that voltage cycles at.	Hz	N/A
Current (I)	The movement of the electric charge through a region.	A	N/A
Real Power (P)	The net transferred energy in one direction.	W	$P = S \times cos(\theta)$
Reactive Power (Q)	The rate at which the power is stored and released back by components such as capacitors and inductors.	VAR	$Q = S \times sin(\theta)$
Apparent Power (S)	The combination of voltage and current.	VA	$S = \sqrt{P^2 + Q^2}$
Real Energy (Pt)	The real power consumed in a specific time.	Wh	$Pt = P/time$
Reactive Energy (Qt)	The amount of reactive power in a specific time.	VARh	$Qt = Q/time$
Apparent Energy (St)	The amount of apparent power consumed in a specific time.	VAh	$St = S/time$
Displacement Power Factor (dPF)	The cosine of phase angles between the current and voltage	ratio	$dPF = cos(\theta)$
Apparent Power Factor (aPF)	The ratio of real power to apparent power.	ratio	$aPF = P/S$

- Weather Data: This includes temperature, wind speed, air density and solar radiation.
- DER Related Data: different types of distributed energy resources have different parameters that can influence the amount of energy that is generated. For example, solar panel output can be influenced by four parameters. These include the DC rating, array type, orientation of the panels on the rooftop, and the DC to AC derate factor. Whereas wind turbines are characterised by the following parameters: rotor diameter, swept area of blades and hub height.
- Users' Contextual Data: These are static information about the customer taken at the time of the registration and include: property type, property age, number of tenants, floor area and many more.

Taking into account the drawbacks of the available public datasets and the need for including data from multiple data sources, one solution to consider is the use of simulators to generate a complete set of multi-source data that can be used to evaluate energy theft detection solutions. There exist different types of energy simulation tools ranging from small appliance simulators to whole energy grid system

simulators. Moreover, various programs differ in terms of flexibility and capabilities and hence, it is important that the researcher studies these capabilities and be aware of the limitations in order to select the most appropriate simulator for their intended objective. To be able to study the issue of energy theft attacks in a modernised smart grid system with full regard to all possible scenarios, we take into account the following criteria in choosing the simulator:

- The simulator should simulate end users' consumption usage.
- The simulator should allow end users to use on-site generations and should simulate generation profiles.
- The simulations should be dynamically influenced by weather data.
- The simulations should be influenced by static contextual data such as the floor area of the residency and location.
- The simulator should be able to report different electricity data from each smart meter.

According to the above criteria, we have reviewed six of the most well-known simulators in the smart grid's community. These simulators are listed in Table 2.7 along with the criteria that they fulfil. Another important criterion that we needed to investigate in these simulators is their ability to report different electricity parameters and not only the consumed or generated real power. Table 2.8 lists out the electrical parameters (mentioned in Table 2.6) that these simulators report.

Based on our review of the above criteria, we can say that GridLAB-D is the most comprehensive simulator that can provide all of the necessary functionalities for studying energy theft attacks.

2.10 Discussion and Open Problems

In this chapter, we analysed the current state-of-the-art energy theft detections in terms of the detection techniques used, evaluation metrics and datasets. We specifically analysed 32 recently published work that uses both ML-based and non-ML-based detection techniques. Table 2.9 lists a summary of these detection research work and their properties. As can be observed, we noticed that none have studied the impact of having distributed energy resources on customer premises and being a prosumer on energy thefts. This is because the prosumer concept has only

Table 2.7: *Comparative Table on Available Energy Grid Simulators*

Simulator	End User Load Simulation	Prosumers	Meter's Multi-Dimensional Electricity Data	Weather	Coverage
GridLAB-D [85]	✓	✓	✓	✓	Whole smart grid
RAPSim [86]	✓	✓	✓	✓	Microgrids
OpenDSS [87]	✗	✗	✓	✗	Whole smart grid
SRLS [88]	✓	✓	✗	✓	Residential buildings
LoadProfileGenerator [89]	✓	✓	✗	✓	Residential buildings
EnergyPlus [90]	✓	✓	✓	✓	Commercial & residential buildings

now started to play a leading role in the energy sector [91]. This new actor creates new challenges with regard to detecting energy theft in electricity systems. Current prosumers report the amount that they consume from the grid and the surplus amount that they inject into the grid. With these figures reported, a prosumer can report fake figures in order to steal electricity from the grid or steal money.

Moreover, most of the existing work for detecting energy theft does not take into consideration data features from different sources. Research should exploit the possibility of using different electricity parameters reported by the smart meter (other than consumed real power) to detect abnormalities and also, the possibility of using multiple data sources. Much research aimed at detecting energy thefts using machine learning uses defined datasets, which limits the variability of features that can be included in the detection.

Another challenge that faces energy theft detection is customers' privacy. It was noted that most of the existing detection methods access users' raw energy data without any concerns for their privacy. However, many concerns have been raised by customers as the disclosure of their real-time and fine-grained power consumption can reveal personal private information [11, 24, 92]. This introduces a challenge to propose a privacy-preserving detection technique. However, research on this subject is still very limited, especially using ML-based techniques [2, 10]. In addition, privacy needs to be evaluated and quantified; yet, this was overlooked by researchers in the field. Therefore, we regard addressing it in our proposed work.

Table 2.8: *Electrical Parameters of the Load Simulators*

Simulator	Electrical Parameter*										
	V	**f**	**I**	**dPF**	**aPF**	**P**	**Q**	**S**	**Pt**	**Qt**	**St**
GridLAB-D	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-
RAPSim	✓	-	✓	-	-	✓	✓	-	✓	-	-
OpenDSS	✓	✓	✓	-	✓	✓	✓	-	✓	-	-
SRLS	✓	-	✓	-	-	✓	-	-	✓	-	-
LoadProfileGenerator	-	-	-	-	-	-	-	✓	✓	-	-
EnergyPlus	✓	✓	✓	-	-	✓	-	-	✓**	-	-

* **V:** Voltage; **f:** Frequency; **I:** Current; **dPF:** Displacement power factor; **aPF:** Apparent power factor; **P:** Real power; **Q:** Reactive power; **S:** Apparent power; **Pt:** Real energy; **Qt:** Reactive energy; and **St:** Apparent energy.
** in Joules.

Lastly, an important aspect which is usually overlooked in energy theft detection is the post-detection part. The focus of energy theft detection solutions has only been on the detection part, but it is important to take action beyond that. This includes determining the amount of stolen energy and incorporating it into future energy demand forecasting.

Table 2.9: *Summary of Energy Theft Detection Research Work*

Reference	Energy Theft Detection Approach	Privacy Preserving Approach	Supported Features								
			F1*	F2*	F3*	F4*	F5*	F6*	F7*	F8*	F9*
Cárdenas et al. [31]	Game Theory	N/A	✗[a]	-	✗	✗	✗	✓	✗	✗	✗
Amin et al. [32]	Game Theory	N/A	✗	-	✗	✓	✗	✗	✗	✗	✗
Wei et al. [33]	Game Theory	N/A	✗	-	✗	✓	✗	✗	✗	✓	✗
Grewal et al. [36]	Hardware-Based	N/A	✗	-	✗	✓	✓	✓	✗	✗	✗
Saad et al. [37]	Hardware-Based	N/A	✗	-	✗	✓	✓	✓	✗	✗	✗
Sathyapriya and Jeyalakshmi [38]	Hardware-Based	N/A	✗	-	✗	✓	✓	✓	✗	✗	✗
Gill et al. [39]	Hardware-Based	N/A	✗	-	✗	✓	✓	✓	✗	✗	✗
Huang et al. [41]	State Estimation	N/A	✗	-	✗	✓	✓	✗	✗	✗	✗
Su et al. [42]	State Estimation	N/A	✗	-	✗	✓	✓	✗	✗	✗	✗
Tariq and Poor [43]	State Estimation	N/A	✗[b]	-	✗	✗	✓	✓	✗	✗	✗

Continued on next page

Table 2.9: *Summary of Energy Theft Detection Research Work (Continued)*

Reference	Energy Theft Detection Approach	Privacy Preserving Approach	Supported Features								
			F1*	F2*	F3*	F4*	F5*	F6*	F7*	F8*	F9*
Salinas et al. [44]	State Estimation	Decomposition Algorithms	✓	✗	✗	✓	✗	✗	✗	✓	✗
Salinas and Li [45]	State Estimation	Decomposed Kalman Filter	✓	✗**	✗	✓	✓	✗	✗	✓	✗
Wen et al. [47]	State Estimation	NTRU Cryptosystem	✓	✗**	✗	✓	✓	✗	✗	✓	✗
Gunturi and Sarkar [49]	Supervised ML (Ensemble Learning)	N/A	✗	-	✗	✓	✗	✗	✗	✓	✗
Buzau et al. [50]	Supervised ML (XGBoost)	N/A	✗	-	✗	✓	✓	✗	✓	✗	✗
Yan and Wen [51]	Supervised ML (XGBoost)	N/A	✗	-	✗	✓	✓[c]	✗	✗	✓	✗
Punmiya and Choe [52]	Supervised ML (XGBoost, CatBoost, LlightGBM)	N/A	✗	-	✗	✓	✓[c]	✗	✗	✓	✗
Zheng et al. [53]	Supervised ML (Wide and Deep CNN)	N/A	✗	-	✗	✓	✗	✗	✗	✗	✗
Hasan et al. [54]	Supervised ML (CNN+LSTM)	N/A	✗	-	✗	✓	✗	✗	✗	✗	✗
Hu et al. [55]	Semisupervised ML (FENs+DAE)	N/A	✗[b]	-	✗	✓	✓[d]	✗	✗	✓	✗
Zanetti et al. [56]	Unsupervised ML (FCM)	N/A	✗[b]	-	✗	✓	✓[c,d]	✗	✗	✓	✗
Zheng et al. [57]	Unsupervised ML (Density Cluster)	N/A	✗	-	✗	✗	✗	✗	✗	✓	✗
Singh et al. [58]	Unsupervised ML (PCA)	N/A	✗	-	✗	✓	✓[d]	✗	✗	✓	✗
Richardson et al. [64]	Unsupervised ML (Clustering)	Paillier Cryptosystem	✓	✗	✓	✓	✓	✓	✗	✗	✗
Yao et al. [65]	Supervised ML (CNN)	Paillier Cryptosystem	✓	✗**	✗	✓	✗[d]	✗	✗	✗	✗
Nabil et al. [66]	Supervised ML (CNN)	Secure Multiparty Computation	✓	✗**	✗	✓	✗[c]	✗	✗	✓	✗
Ibrahem et al. [67]	Supervised ML (FNN)	Functional Encryption	✓	✗**	✗	✓	✗[d]	✗	✗	✓	✗
Wen et al. [68]	Supervised ML (TCN)	Federated Learning and Local Differential Privacy	✓	✗**	✗	✓	✗	✗	✗	✗	✗
Ashraf et al. [69]	Supervised ML (Ensemble Learning)	Federated Learning	✓	✗	✗	✓	✗	✗	✗	✗	✗

Continued on next page

Table 2.9: *Summary of Energy Theft Detection Research Work (Continued)*

Reference	Energy Theft Detection Approach	Privacy Preserving Approach	Supported Features								
			F1*	F2*	F3*	F4*	F5*	F6*	F7*	F8*	F9*
Muzumdar et al. [73]	Difference between energy supply and consumption	Bloackchain	✓	✗**	✗	✓	✓	✗	✗	✗	✗
Messinis et al. [74]	Hybrid Solution (State Estimation + Supervised ML)	N/A	✗	-	✗	✗	✗	✗	✓	✓	✗
de Souza et al. [75]	Hybrid Solution (State Estimation + Supervised ML)	N/A	✗[b]	-	✗	✓	✓	✓	✓	✗	✗

* **F1:** Privacy preservation; **F2:** Privacy quantitative analysis; **F3:** Detecting prosumers' thefts; **F4:** Pinpointing a thief; **F5:** Pinpointing time of theft; **F6:** No requirement for historical data; **F7:** Usage of multi-source data; **F8:** Detecting multiple energy thefts; and **F9:** Considering demand-response management after the detection.
** The study only provides qualitative privacy analysis, not a quantitative one.
[a] Only at the demand model.
[b] Only assumed from reducing the frequency of the readings.
[c] Only the day of stealing is identified.
[d] Only the week of stealing is identified.

2.11 Research Model

In this thesis, we have conducted three empirical studies that complement each other for developing energy theft detectors. In these studies, we took into account the aforementioned limitations and problems of the current literature. Each study addresses one of the following research hypotheses in a separate chapter:

- ***Hypothesis 1:*** *Combining machine learning techniques (clustering and classification) can enhance the detection of a range of thefts, including prosumers thefts.*

- ***Hypothesis 2:*** *A privacy-preserving ML technique that suits the smart grid environment can be developed to accurately and effectively detect energy theft while preserving the privacy of customers' data.*

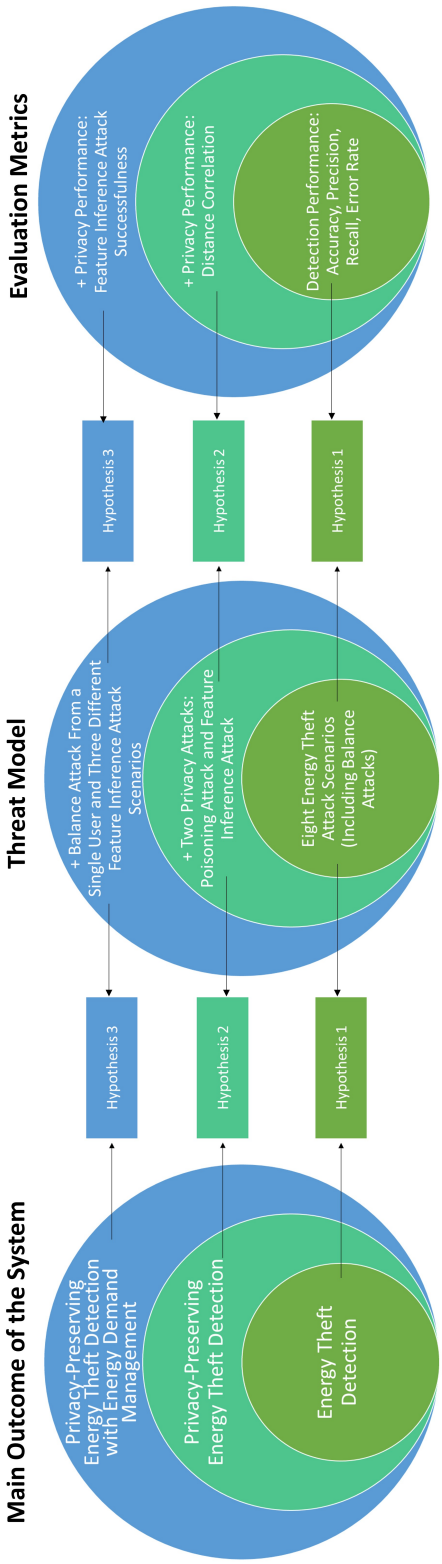
- ***Hypothesis 3:*** *A multi-output neural network framework can be used to simultaneously predict the presence of theft, predict its magnitude, and use that estimation to make more accurate forecasts.*

The *first hypothesis* is addressed in Chapter 3 which develops a cluster-based ML energy theft detection model. In this work, we consider eight different energy theft scenarios as part of our threat model. These scenarios include a new type of attack that we propose which we refer to as *balance attack*. The detection is done in three phases, starting with clustering the users, decomposing the time series and lastly classifying each point as theft or not using different well-known ML classifiers. The performance of the proposed energy theft detector here is evaluated using four evaluation metrics.

Building on this work, we develop a privacy-preserving energy theft detector to address the *second hypothesis* in Chapter 4. In the threat model, we consider the same set of energy theft scenarios as the one proposed in Chapter 3 with additional two privacy attacks: poisoning attack and feature inference attack. As stated in the hypothesis, the detection methodology needs to be privacy-aware, and therefore we propose a new variant of an ML architecture called *three-tier split learning* that suits the nature of smart grids. The proposed model uses a stacked auto-encoder as the underlying detection methodology. Adding privacy on top of energy theft detection requires us to evaluate the privacy gain of the proposed model. Hence, we use a metric called *distance correlation* to evaluate the privacy aspect of the model.

For our *third and last hypothesis*, we develop a multi-output neural network and enhanced-privacy preserving model that uses a masking approach and a noisy layer neural network to evaluate the hypothesis. We show how detecting thefts and estimating their magnitudes can actually help in estimating future demand. Our threat model here is expanded to include a more comprehensive set of feature inference attacks along with another privacy metric to evaluate the success level of these attacks.

The way these hypotheses are addressed by our proposed studies, and how the evolution of these studies is handled, can be seen in Figure 2.6.



Detection Methodology

Hypothesis 1	Hypothesis 2	Hypothesis 3
Clustering + Timeseries Decomposition + Multiple Supervised ML Approaches	Three Tier Split Learning + Stacked Auto-Encoder	Three Tier Split Learning + Multi-Output Neural Network
Input size is one timestamp	Input size is 16 timestamps	Input size is one timestamp

Figure 2.6: *Research Model Showing the Relationships Between the Thesis's Studies*

Chapter 3

ML-Based Detection Model in the Presence of Prosumers

Data-driven approaches have been widely employed in recent years to detect energy thefts. Although many techniques have been proposed in the literature, they mainly focus on energy thefts by power grid consumers. Existing studies do not consider energy thefts by *prosumers*, who act as both producers and consumers in the energy system. This is of great importance as inaccurate reports of prosumers' behaviours can disrupt power system operations. This chapter examines the prosumers' role in subverting the energy system and proposes a novel means of detecting such malfeasance. Moreover, we introduce *new* energy theft attack scenarios called *balance attacks*, where an attacker concurrently modifies his readings along with neighbouring meters in an attempt to balance the total aggregated reading. Such attacks can be difficult to detect by existing solutions that reach detection decisions based on aggregated readings. Existing approaches use either a single model for all users across the system or else a model for each user. Here, we adopt a halfway house approach and propose a cluster-based detection model. For users in a cluster, we decompose the power time series data into trend, cyclical and residual components. Residual data, along with different features from multiple data sources, are fed into an ML classification algorithm to detect anomalous readings. Simulations have been conducted using a newly generated dataset, and results have shown that the proposed model can detect energy theft with high detection and low error rates. The results also show that the model can detect thefts by new users with great accuracy.

3.1 Introduction

Recently, the incorporation of distributed energy resources (DERs) in a user's premises, allowing a user to generate, store and supply electricity, has received significant attention. Here, the stakeholder is generally called a "prosumer" (producer-consumer). This actor was overlooked by scholars in the energy theft research area since it is a new participant in the grid. However, considering the role of prosumers is important as their number is increasing rapidly; according to the European Renewable Energies Federation [93], the UK had almost 1 million prosumers in 2015 and will likely have 24 million by 2050. Prosumer theft can be carried out by manipulating consumption and generation data; and as pointed out in Section 2.10, the existing research has not studied this impact (represented as F3 in Table 2.9). Prosumers are different from traditional consumers as they not only use energy but also generate and store or transfer surplus energy to the grid. This allows malicious prosumers to manipulate data regarding their generation and consumption, which can introduce an imbalance in the overall grids' data. Moreover, prosumer thefts can disrupt the energy supply to a region, cause grid instability or deny energy access to other users in that area [94, 95]. Hence, the detection of prosumers' attacks is of great importance. To manage prosumers, it is critical to understand their generation and consumption behaviours [94]. The analysis of all factors of prosumer behaviour helps to build and plan for the proper balance of energy demand and supply.

Moreover, some existing energy theft detection research do not identify *which* user is the thief (represented as F4 in Table 2.9), and many other detection methods classify each user as either thief or honest but do not identify the time of theft (represented as F5 in Table 2.9). Additionally, most recent studies have not availed themselves of data features from different sources (represented as F7 in Table 2.9). Machine learning approaches usually consider a single electrical feature (consumed power), while smart meters report more than ten different electrical parameters [96]. This abundance of unused data is an opportunity.

The last limitation is that most existing solutions use one of two approaches: a generalised model or a user-specific model. In the generalised model, a single honest reference model is created using data from all users. This can result in a detection scheme with low accuracy. On the other hand, user-specific models, where a separate model is developed for each user using their data, can become difficult to scale. Therefore, a cluster-based detection model can be the ideal combination of the two approaches.

3.1.1 Our Contribution

This work addresses all the above limitations where the specific contributions are:

- The *first* theft detection method to be based on the use of user clustering (with reference models built for each cluster) and the first to address theft by both consumers and prosumers. The approach has further desirable properties, e.g., the ability to detect thefts from new users without the need for historical data.
- The introduction of new energy theft scenarios, which we term *balance attacks*, that can balance the amount of electricity stolen at one meter with manipulated values returned from other neighbouring meters. This scenario can be hard to detect by existing detection models.
- The production of a benchmark dataset that includes examples of an extensive range of data injection attacks (including balance attacks).
- An evaluation of the use of various ML techniques for the classification of customers' behaviours.

The rest of this chapter is organised as follows: Section 3.2 provides the system architectural model and the threat model. Section 3.3 describes how the proposed detection system is designed. Sections 3.4 and 3.5 detail the experimental setup and results. Section 3.6 discusses threats to the validity of our study, and how they have been mitigated. Finally, Section 3.7 gives concluding remarks and directions for the next chapter.

3.2 System Model and Threat Model

3.2.1 System Model

We consider a typical smart grid system model, shown in Figure 3.1. It consists of three major entities: a set of clients, a set of substation gateways (GWs), and a server at a control centre. Specifically, each entity has the following roles in the system:

- Clients are homeowners with smart meters that send data to the substation gateways at fixed intervals (e.g. every 15 minutes). A client can be either a

consumer (who consumes electricity from the grid) or a prosumer (who both supplies and consumes electricity to/from the grid). Prosumers generate their own electricity using a dedicated distributed energy resource (DER) such as solar panels or wind turbines. A consumer, i, is equipped with one smart meter that is responsible for collecting energy consumption data (CSM_i), whereas a prosumer j is equipped with an additional smart meter for energy production data (PSM_j).

- Substation gateways are third-party components that facilitate the communication and electricity flow between the control centre and clients. Each gateway is responsible for periodically collecting the energy measurement data of a group of clients in a geographical location, called a neighbourhood area network (NAN), and sending them to the utility server.
- The server is the utility control centre responsible for distributing electricity to all clients. It also uses all system data to manage the electricity demand for the next period and maintain the balance between power generation and demand. Abnormalities in either consumption or generation reports are detected at the server.

3.2.2 Threat Model

Energy theft can be carried out by manipulating the reported energy readings. In our threat model, we allow an adversary (a malicious client) to change their meter readings to pay a lower consumption bill or get paid for electricity that they did not generate. An adversary can manipulate consumption and production readings at any point in the system as shown in Figure 3.1. The adversary can inject false measurements by manipulating the configuration of a smart meter or attacking the communication channels, either physically or through cyber attacks. Therefore, our threat model considers two types of adversaries:

- *An External Adversary:* who may try to tamper with the readings of SMs either physically or through cyber-attacks. The external adversary is also able to intercept readings and change them during communication.
- *An Internal Adversary:* who can be an insider that can change the readings at either the substation or control centre where data resides.

Both external and internal adversaries can modify the meter readings of a consumption SM (CSM) or a production SM (PSM) using different attack scenarios

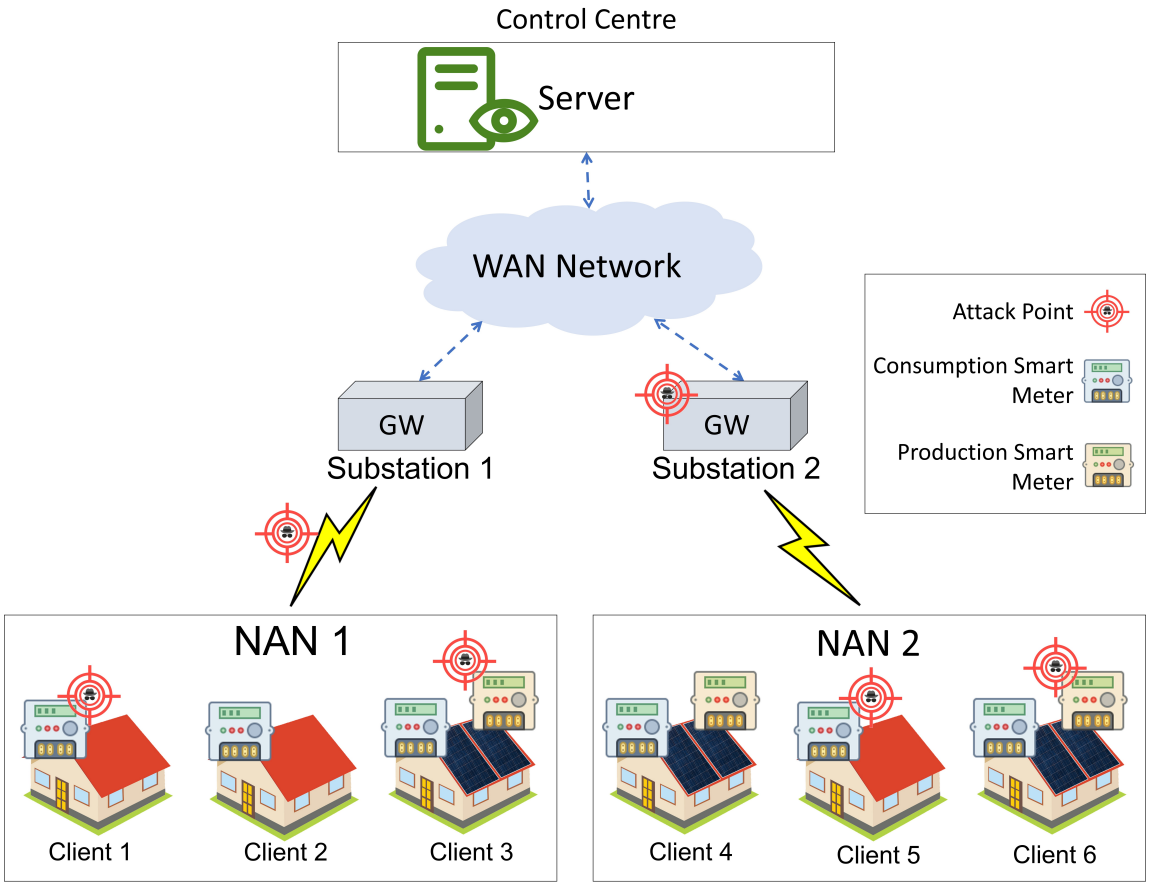


Figure 3.1: *System Model Showing Some of the Possible Attack Points (Not All Attack Points are Shown) [97]*

as listed in Table 3.1. In our threat model, we consider eight different energy theft attack (ETA) scenarios that can be launched by the two adversaries mentioned before. The first four of these attacks have been developed with the help of the widely used mathematical model defined in [46]. The additional four scenarios have considered attacks where one reported consumption is maliciously *increased* to balance a malicious *decrease* in another. By this, the attacker is either stealing from another client (if they have the same tariff) or collaborating with another client to steal from the grid (if they have different tariffs). Our model incorporates such attacks, and we refer to them as *balance attacks*. We assume these attacks can be launched by either a single attacker or collaboratively using collusive attacks. In general, our attacks can be viewed in four categories: consumer thefts, prosumer thefts, consumer balance thefts and prosumer balance thefts. Each attack category is described below and is summarised in Table 3.1.

- In consumer thefts (attack scenarios #1 and #2), a user i (either a consumer or a prosumer) may wish to reduce their consumption smart meter reading CSM_i by either a constant value l or a percentage k for a period of time T.
- In prosumer thefts (attack scenarios #3 and #4), a prosumer i may wish to increase their production smart meter reading PSM_i by a constant value l or a percentage k for a period of time T.
- In consumer balance thefts (attack scenarios #5 and #6), a user i (either consumer or prosumer) may wish to reduce their reported consumption smart meter reading CSM_i by a constant value l or a percentage k for a period of time T and increase another user's reported consumption CSM_j by the same energy amount. This is done to maintain the total energy consumed and reported by the two users. We refer to this as a "balance attack".
- In prosumer balance thefts (attack scenarios #7 and #8), a prosumer i may wish to increase their production smart meter reading PSM_i by a constant value l or a percentage k for a period of time T and decrease another prosumer's reported production PSM_j by the same energy amount. This is done to maintain the total energy produced and reported by the two users. This is also a balance attack in terms of production.

3.3 Proposed Detection Model

Figure 3.2 shows the three phases of our proposed detection approach. The phases are described below.

Table 3.1: *Overview of Attack Scenarios*

Attack Type		Attack Scenario
Consumers Thefts	Attack #1	$CSM'_i = CSM_i - l$
	Attack #2	$CSM'_i = CSM_i - (CSM_i \times k/100)$
Prosumers Thefts	Attack #3	$PSM'_i = PSM_i + l$
	Attack #4	$PSM'_i = PSM_i + (PSM_i \times k/100)$
Consumers Balance Thefts	Attack #5	$CSM'_i = CSM_i - l$ and $CSM'_j = CSM_j + l$
	Attack #6	$CSM'_i = CSM_i - (CSM_i \times k/100)$ and $CSM'_j = CSM_j + (CSM_i \times k/100)$
Prosumers Balance Thefts	Attack #7	$PSM'_i = PSM_i + l$ and $PSM'_j = PSM_j - l$
	Attack #8	$PSM'_i = PSM_i + (PSM_i \times k/100)$ and $PSM'_j = PSM_j - (PSM_i \times k/100)$

Balance Attacks

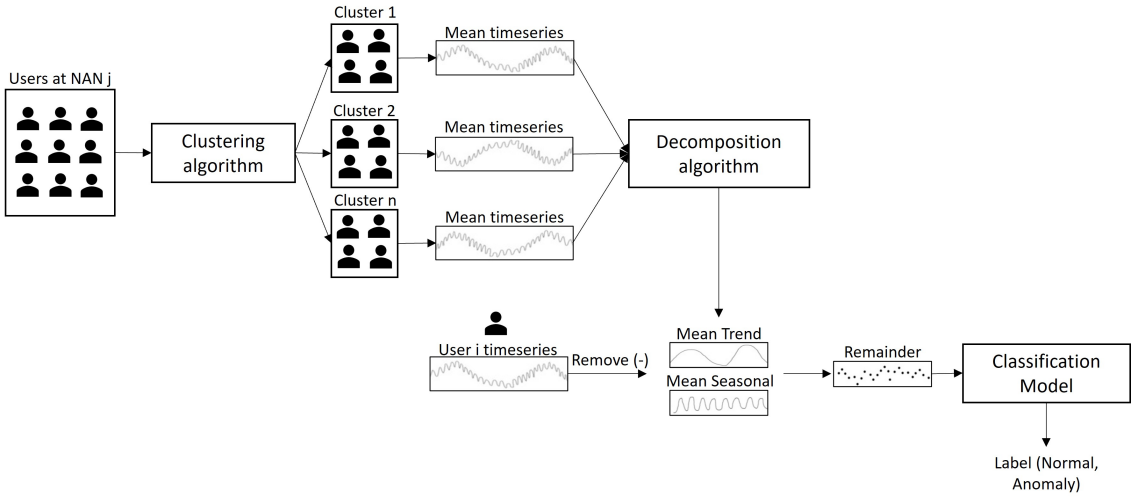


Figure 3.2: *Overview of the Detection System*

3.3.1 User Clustering

Two detection mechanisms have been proposed by researchers: generalised models and user-specific models [98]. Generalised detection models, as illustrated in Figure 3.3a, are built using all users' data, meaning a single model for all users is used to detect energy thefts. User-specific detection models are specific models that are built using only the dataset of that user, which means that a system has n models for n users as shown in Figure 3.3c. Generalised models have the advantage of detecting thefts by new users; however, as these models use the average of all users, they might suffer from low accuracy. User-specific models are generally more accurate but encounter significant scaling issues. Our approach offers a halfway house: it clusters users and develops a reference model for each cluster.

Users can be clustered based on their electricity consumption profiles. This approach has already been used in [57] and [99]. Another option is to cluster the users based on their geographical location and user residence characteristics. We have adopted this approach. Users who share the same geographical location and residence physical characteristics are likely to have a similar pattern of consumption and generation. According to Eurostat [100], people in the same neighbourhood are more likely to have similar incomes, which in turn, affects the physical characteristics of their building and the types of equipment and appliances that they use. Hence, their consumption and generation patterns will typically be similar, whilst users in different clusters can have different usage and generation profiles. Therefore, Our first phase of the proposed model is to cluster the users using 14 static features (reported

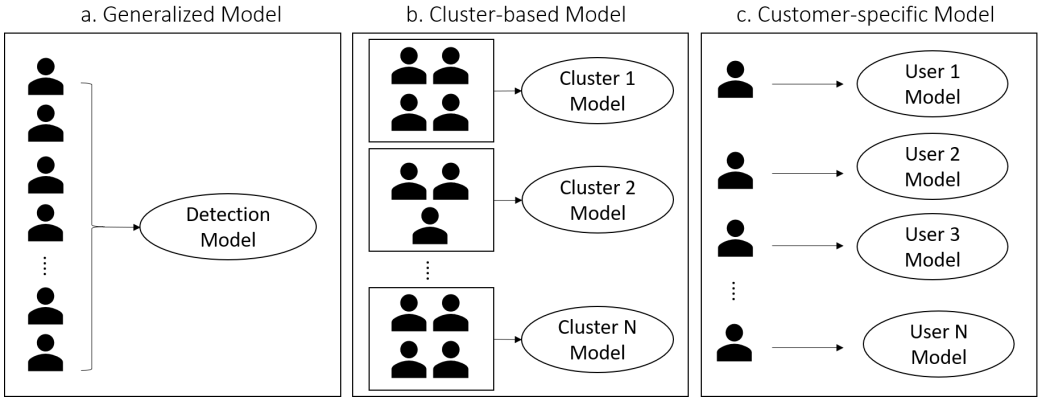


Figure 3.3: *Detection Model Types*

in Table 3.2) representing the residential characteristics of a client's property. We start by first preprocessing the feature set and then clustering them to partition users in each NAN. This has been tested using different unsupervised clustering algorithms: K-means, hierarchical clustering, density-based spatial clustering and agglomerative clustering. Clusters were then visualised and two of the best clustering algorithms that have been applied can be seen in Figure 3.4. From the figure, it can be seen that *agglomerative clustering* creates more distinct clusters than K-means. The number of clusters in each NAN is chosen based on minimising the total within-cluster sum of square (WSS) (Elbow method). The data from each cluster is then processed individually in the next phase.

3.3.2 Time-series Decomposition

We assume that energy theft points are identified as outliers from usual behaviour time-series data. Finding outliers in time-series data can be done using time-series decomposition. A time-series data Y at time t is composed of three components: a trend component T_t, a seasonal/cyclical component S_t and a residual (remainder) component R_t. These components are either added or multiplied together to form the original signal as follows:

$$Y_t = T_t * S_t * R_t \quad or \quad Y_t = T_t + S_t + R_t \tag{3.1}$$

To automatically decompose a time series into its components, different methods have been proposed, such as seasonal-trend decomposition using regression (STR) [101], singular spectrum analysis (SSA) [102], or decomposition of time series

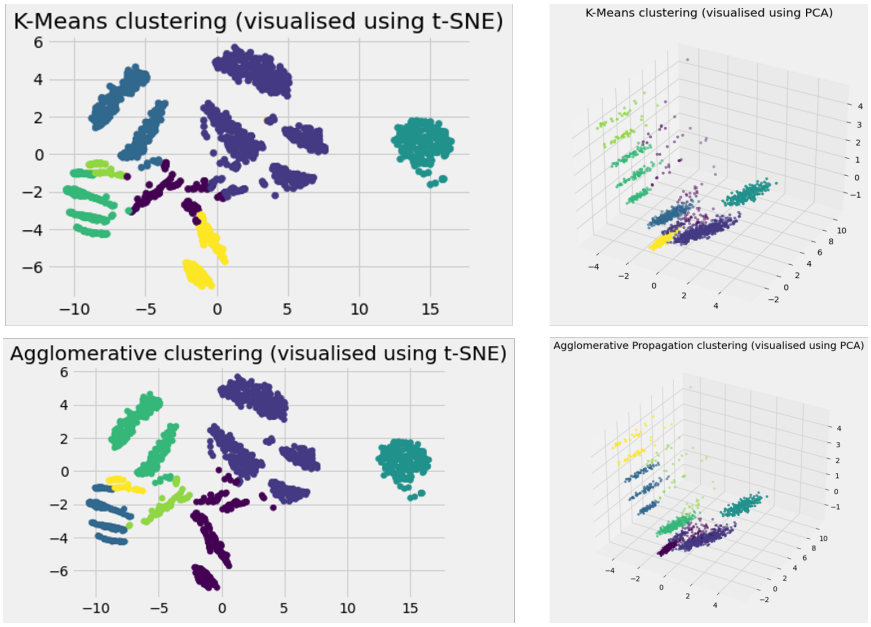


Figure 3.4: *Clustering of Customers Using Agglomerative and K-means Clustering Methods*

by Loess (STL) [103]. In this work, the additive STL decomposition method is used to decompose the users' consumption and generation series. This is because STL can handle any type of seasonality/cyclicity. In particular, it can handle the daily cycle apparent in our data. For each cluster, the average consumption/generation of all users is computed and then decomposed to obtain the cyclical (here daily) and trend components (see Figure 3.5). For each user, those components are removed from his/her data to obtain the residuals. We found that obtaining the residuals from removing the cluster's trend and daily components creates more distance between normal and anomalous data points than removing all users' trend and daily components. This increase in distance, as shown in Figure 3.6, creates a separation between normal and anomalous data.

3.3.3 Classification

Our final phase classifies a vector of 15 features (both dynamic and weather features reported in Table 3.2) as either an anomaly or a normal point. In the training phase, a balanced dataset of equally normal and anomalous data points is used. The dataset involves multiple features along with the residuals obtained from the previous phase. As each data point consists of a vector of features with different value ranges, these features are first normalised using the StandardScalar [104]. Different machine

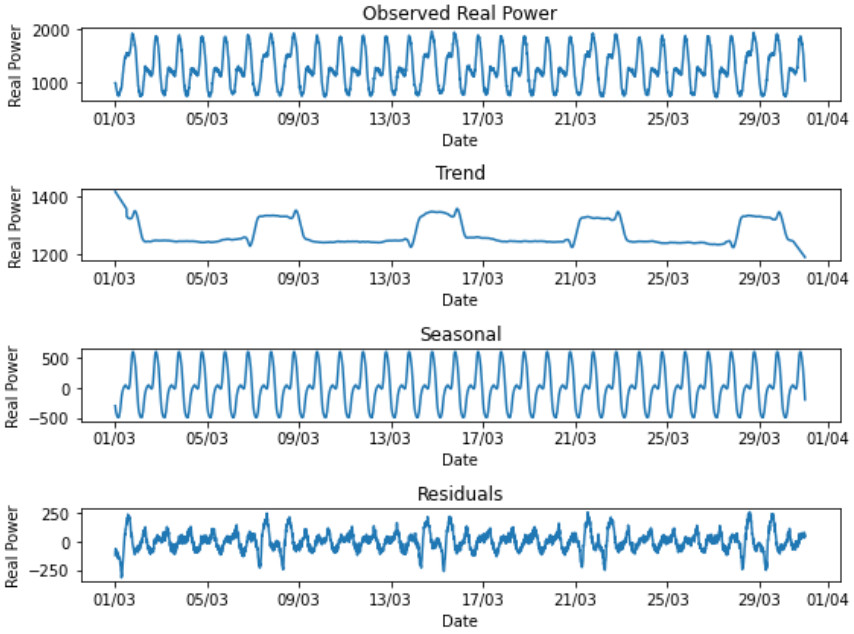


Figure 3.5: *Decomposition of Consumed Power for a Cluster*

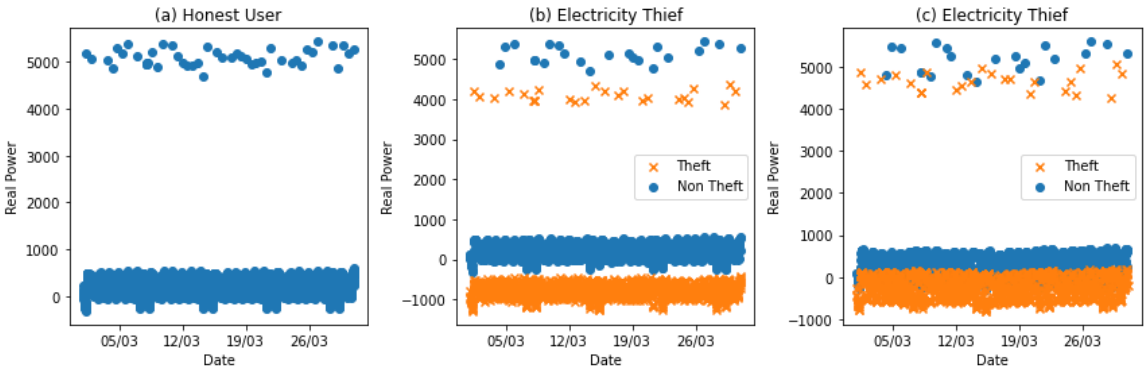


Figure 3.6: *Residuals of (a) Honest User, (b) Electricity Thief After Removing the Trend and Daily Components of Cluster's Users, and (c) Electricity Thief After Removing the Trend and Daily Components of All Users*

learning algorithms are then applied to this normalised data to make the decision. After training the model, the model is used to detect data thefts in unseen data points. At the end of this phase, the system will have a single detection model for each cluster of users.

3.4 Experimental Setup

3.4.1 Dataset Generation

In order to study the energy theft scenarios that are carried out by both consumers and prosumers, we need a dataset that includes the electricity usage of users from both types. In Chapter 2, we have reviewed the public energy usage datasets used by the current literature. The review showed that these datasets lack some important features that we require in our proposed model. None has provided energy usage records for both consumers and prosumers. The datasets also lack contextual data that might affect the consumption and generation of a user, e.g. the floor area of the residency, its location and prevailing weather conditions.

Due to these limitations, we have generated our own dataset using "GridLab-D" [85], which is a powerful simulation tool that simulates power flows between the grid's entities. GridLab-D is very flexible as it allows reporting both production and consumption data that are dynamically influenced by weather data. It was chosen since it achieves all the necessary criteria, listed in Section 2.9, that help in studying different scenarios of energy theft attacks.

We used the taxonomy distribution feeder, R1-12.47-2, developed by Pacific Northwest National Laboratory (PNNL) [105] to produce a detailed distribution feeder model in GridLab-D format for use in generating the dataset for our work. This distribution feeder represents a moderately populated suburban and rural area composed of 1594 residential users with varying loads and physical properties, where 49 of those users are prosumers with solar panels. Our dataset not only contains consumption and generation profiles of both consumers and prosumers, but also reports multiple electrical parameters every 15 minutes. It contains weather conditions and users' static residence characteristics. The script provided by PNNL has been modified to allow the reporting of all features listed in Table 3.2 every 15 minutes for every user. To allow research reproducibility, the original dataset (without any injection of attacks) along with a description of each feature has been published in our GitHub repository[1].

[1] https://github.com/asr-vip/Electricity-Theft

Table 3.2: *Features of the Dataset*

	Feature	Description	Unit	Data Type
Static Parameters	Floor area	Home conditioned floor area	sf	Integer
	Stories	Number of stories in the home	-	Integer
	Ceiling height	Average ceiling height	ft	Integer
	Roof's R-value	A value that represents the effectiveness of insulating material and heat flow across the roofs	degF.sf.h/Btu	Float
	Wall's R-value	A value that represents the effectiveness of insulating material and heat flow across the walls	degF.sf.h/Btu	Float
	Floor's R-value	A value that represents the effectiveness of insulating material and heat flow across the floors	degF.sf.h/Btu	Float
	Door's R-value	A value that represents the effectiveness of insulating material and heat flow across the doors	degF.sf.h/Btu	Float
	Glazing layers	Number of glass layers in each window	-	String
	Glass type	The type of window glass used (LOW_E_GLASS, GLASS, OTHER)	-	Category
	Glazing treatment	The treatment type used for exterior windows (HIGH_S, LOW_S, REFL, ABS, CLEAR, OTHER)	-	Category
	Window frame	The type of window frame (INSULATED, WOOD, THERMAL_BREAK, ALUMINUM, NONE)	-	Category
	Heating system	Heating mechanism for house (RESISTANCE, HEAT_PUMP, GAS, NONE)	-	Category
	Cooling system	Cooling mechanism for house (HEAT_PUMP, ELECTRIC, NONE)	-	Category
	Solar panel size	Area of the solar panel	ft	Integer

Continued on next page

Table 3.2: *Features of the Dataset (Continued)*

	Feature	Description	Unit	Data Type
Dynamic Parameters	Real power	Measurement of the consumed real portion of the power flowing through the meter at time stamp	watt	Float
	Voltage	Measurement of the voltage of the meter	volts	Float
	Real energy	Measurement of the real energy (accumulation of the real power) that has flowed through the meter	watt-hour	Float
	Reactive power	Measurement of the reactive portion of the power flowing through the meter at a single timestamp	volt-amperes	Float
	Reactive energy	Measurement of the reactive energy (accumulation of the reactive power) that has flowed through the meter	VA-hours	Float
	Current	Measurement of the current of the meter at a single timestamp	amperes	Float
	Apparent power	Measurement of the apparent power (active power + reactive power) that has flowed through the meter	volt-amperes	Float
	Solar Value	Measurement of the generated power generated by the solar panel at a single timestamp	watt	Float
Weather Parameters	Temperature (Dry-Bulb)	Dry bulb temperature at the time indicated	deg C	Float
	Pressure	Station pressure at the time indicated	mbar	Float
	RHum	Relatitudeive humidity at the time indicated	percent	Float
	TotCld	Amount of sky dome covered by clouds or obscuring phenomena at time stamp	tenths of sky	Float
	GHI	Direct and diffuse horizontal radiation received during 60 minutes prior to timestamp	Wh/m^2	Float

Continued on next page

Table 3.2: *Features of the Dataset (Continued)*

	Feature	Description	Unit	Data Type
Weather Parameters	ETR	Extraterrestrial horizontal radiation received during 60 minutes prior to timestamp	Wh/m^2	Float
	GHillum	Avg. total horizontal illuminance received during the 60 minutes prior to timestamp	lx	float
	Zenithlum	Avg. luminance at the sky's zenith during the 60 minutes prior to timestamp	cd/m^2	Float
	Wx	The x component of wind direction and wind speed at the time indicated. This is calculated as Wx= wind speed *cos(wind direction in radian)	-	Float
	Wy	The y component of wind direction and wind speed at the time indicated. This is calculated as Wy=wind speed *sin(wind direction in radian)	-	Float

3.4.2 Attack Modes Simulation

We modified the generated profiles in our dataset as they contain only honest (real) readings and implemented the set of theft scenarios that were considered in our attack model. This practice in energy theft detection research, where data theft scenarios are synthetically created and used for training and evaluating the detection model, was first presented in [46] and is now a common practice in the research literature [49, 51, 52, 55, 56, 57, 58, 66, 67, 74, 75].

We created 9 different datasets: one for every attack scenario presented in the threat model and one dataset with all 8 attacks combined. Each attack dataset is balanced where one-half of the readings are theft and the other half are benign. The combined dataset is also balanced with each attack type making up approximately 1/8 of the combined attacks. In our experiments, the values of l and k, defined in the attack scenarios in Table 3.1, were set to 500 and 40 respectively.

3.4.3 Simulation Environment

The proposed detection model is tested using several well-known benchmark ML algorithms which include: decision tree (DT), K-nearest neighbour (KNN), logistic regression (LR), Naive Bayes (NB), multilayer perceptron neural network (NN), and support vector machine (SVM). These ML algorithms were trained and tested using scikit-learn [104] in the Anaconda3 environment using Python [106]. For each ML algorithm, we used the default hyper-parameters provided by scikit-learn. These hyper-parameters are provided in Table 3.3.

For preprocessing the dataset features, we used one-hot encoding for the categorical features and normalised numerical features using the StandardScalar [104] technique. One-hot encoding allows machine learning methods to treat categorical features as numerical ones [107]. The use of one-hot encoding rather than other encodings, such as label encoding, helps in highlighting the presence/absence of features rather than introducing artificial ordering among the categories [108]. The dataset is split into 80% training and 20% testing where the model is trained with 10-fold cross-validation with the underlying optimization score being accuracy. In the first phase, *clustering* phase, the set of static parameters shown in Table 3.2 are used, whereas the remaining set of parameters, along with the residuals from the second phase, are used in the *classification* phase.

3.4.4 Evaluation Metrics

As discussed in Section 2.8, several metrics can be used to evaluate an energy theft detector. Here, we use accuracy, recall (also known as detection rate (DR)), precision and error rate to report our results. Other metrics such as F1-score can be easily calculated from the reported metrics. Our motivation is to obtain high accuracy and recall (detection rate) with a low error rate. In this work, theft data points are denoted as the positive class and benign data points as the negative class.

In Section 3.4.3, we explain that the evaluation used to derive the results is by maximising accuracy in the 10-fold cross-validation. The above indicates what evaluation metrics are used to report the performance of the final ML models obtained.

Table 3.3: *The Default Hyper-parameters of Scikit-learn Classifiers*

ML Model[a]	Default Hyper-parameters
DT	criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0
KNN	n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None
LR	penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None
NB	priors=None, var_smoothing=1e-09
NN	hidden_layer_sizes=(100,), activation='relu', solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000
SVM	C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=None

[a] **DT**: Decision Tree; **KNN**: k-Nearest Neighbors; **LR**: Logistic Regression; **NB**: Naive Bayes; **NN**: Neural Network; and **SVM**: Support Vector Machine.

3.5 Results and Discussion

We evaluate the following aspects of our detection system:

- Impact of different types of attacks.
- Impact of using clustering and time series decomposition on detection performance.
- Detecting theft from new users.
- Impact of changing the percentage of thieves within a cluster.
- Impact of stealing different magnitudes of electricity.

These points are discussed in the following subsections.

3.5.1 Impact of Different Attacks

We tested the overall detection performance for each of the attack scenarios discussed in Section 3.2.2. Table 3.4 shows the accuracy, recall (detection rate), precision and error rate of these different attack scenarios. As indicated above, the results reported are the average of 10-fold cross-validation over a balanced dataset. We also tested the detection of attacks in the combined dataset.

The results in Table 3.4 show that our detection model has a good performance in detecting all attack types. From Table 3.4, attacks #1 #2, #3 and #4 are detected with a detection rate of 99.9%, 99.7%, 99.9% and 98.1% respectively using a neural network ML model. Attacks #5, #6, #7 and #8 are detected with a detection rate of above 92% using a decision tree ML model. In the combined dataset *AllAttacks*, our NN-based detection model can detect any attack type with a detection rate of 96.7%. These results show that the proposed model can detect different attacks with high detection probability when using either the DT or NN classifiers. We can also see that the model performance in attacks #5, #6, #7 and #8 is lower than the other types which indicates that balance attacks can be slightly more difficult to detect, perhaps due to the intrinsic property of zero overall theft. This is very evident in cases of LR and SVM ML models where the two models exhibit similar performance. This supports the claims given by [109] which states that the results of the two ML models are often the same because of the similarities between their loss functions.

Table 3.4: *Experimental Results of the Proposed Model Under Different Attacks*

Metric	Method*	Attack Scenario								
		#1	#2	#3	#4	#5	#6	#7	#8	All Attacks
Accuracy	**DT**	0.998	0.991	0.997	0.978	0.993	**0.950**	0.989	**0.928**	0.953
	KNN	0.981	0.961	0.959	0.887	0.985	0.878	0.975	0.867	0.842
	LR	0.958	0.922	0.578	0.508	0.962	0.846	0.520	0.515	0.768
	NB	0.880	0.716	0.711	0.607	0.923	0.820	0.859	0.788	0.579
	NN	**0.999**	**0.995**	**0.999**	**0.986**	**0.994**	0.906	**0.991**	0.885	**0.973**
	SVM	0.953	0.921	0.577	0.507	0.961	0.846	0.523	0.518	0.770
Recall (DR)	**DT**	0.998	0.991	0.997	0.977	0.993	**0.949**	0.988	**0.925**	0.952
	KNN	0.967	0.937	0.927	0.798	0.976	0.809	0.954	0.780	0.726
	LR	0.954	0.925	0.493	0.367	0.970	0.800	0.255	0.252	0.720
	NB	0.951	0.962	0.478	0.416	0.914	0.714	0.833	0.691	0.447
	NN	**0.999**	**0.997**	**0.999**	**0.981**	**0.995**	0.900	**0.990**	0.857	**0.967**
	SVM	0.949	0.925	0.494	0.380	0.970	0.792	0.261	0.269	0.718
Precision	**DT**	0.998	0.990	0.997	0.978	0.993	**0.950**	0.989	0.929	0.954
	KNN	0.994	0.983	0.990	0.965	**0.994**	0.939	**0.994**	**0.940**	0.945
	LR	0.962	0.918	0.601	0.513	0.954	0.882	0.521	0.515	0.797
	NB	0.853	0.671	0.822	0.723	0.931	0.905	0.874	0.850	0.597
	NN	**0.999**	**0.993**	**0.999**	**0.990**	**0.994**	0.911	0.991	0.905	**0.980**
	SVM	0.957	0.917	0.598	0.510	0.954	0.888	0.527	0.520	0.799
Error Rate	**DT**	0.002	0.009	0.003	0.022	0.007	**0.050**	0.011	**0.072**	0.047
	KNN	0.019	0.039	0.041	0.113	0.015	0.122	0.025	0.133	0.158
	LR	0.042	0.078	0.422	0.492	0.038	0.154	0.480	0.485	0.232
	NB	0.120	0.284	0.289	0.393	0.077	0.180	0.141	0.212	0.421
	NN	**0.001**	**0.005**	**0.001**	**0.014**	**0.006**	0.094	**0.009**	0.115	**0.027**
	SVM	0.047	0.079	0.423	0.493	0.039	0.154	0.477	0.482	0.230

* **DT**: Decision Tree; **KNN**: k-Nearest Neighbors; **LR**: Logistic Regression; **NB**: Naive Bayes; **NN**: Neural Network; and **SVM**: Support Vector Machine.

3.5.2 Impact of Using Clustering and Time Series Decomposition

This section compares our proposed model with a simpler version where clustering and time series decomposition are not used. The first model, which we will refer to as the proposed model, implements the three phases presented in Section 3.3. The second model, the non-cluster and non-time-series decomposition model, only implements the third phase, which is classification.

The results of the comparison are shown in Table 3.5. As can be seen from the table, the proposed model with clustering and time-series decomposition outperforms the non-cluster and non-time-series decomposition model on all four metrics. This suggests that the use of clustering and time-series decomposition enhances the detection model performance. This was also reflected in Figure 3.6 where we showed that the use of clustering increases the distance between theft and non-theft data points.

3.5.3 Impact of Thefts From New Users

Here we evaluate our detection model in terms of detecting thefts from new users. We trained the classifier using the combined dataset *AllAttacks* that includes samples of all attack types. After that, we used a test dataset of users that have not been included during the training phase to evaluate our model. Table 3.6 shows how well the detection model works in detecting thefts from new users. We can observe that the best performance in terms of accuracy, recall, precision and error rate was given by the neural network classifier. Our model can detect thefts from new users without the need for historical data with a detection rate of 93.2% and only a 7.1% error rate.

3.5.4 Impact of Different Percentages of Thieves

This setting analyses the effect of the percentage of thieves that exists in a single cluster. As this is an important factor to take into consideration which can show how well the detection algorithm works in cases of low numbers of thieves. We conducted our experiment using the combined dataset *AllAttacks* which contains all attack types. In this setting, we first trained the model using 10-fold cross-validation and then tested the model using a completely unseen and unbalanced dataset.

Table 3.7 shows the results where we randomly changed the percentage of thieves in a cluster to range from 2% to 20%. The results indicate that our method actually

Table 3.5: *Performance of the Detection Model With and Without Clustering and Time-series Decomposition*

Metric	Method*	Proposed Model	Non-cluster and Non-time-series Decomposition Model	Percentage of Improvement
Accuracy	**DT**	0.954	0.802	18.9
	KNN	0.791	0.603	31.2
	LR	0.747	0.605	23.5
	NB	0.634	0.612	3.69
	NN	0.976	0.820	19.1
	SVM	0.742	0.603	23.2
Recall (DR)	**DT**	0.961	0.797	20.6
	KNN	0.622	0.369	68.7
	LR	0.708	0.632	12.0
	NB	0.527	0.857	-38.6
	NN	0.972	0.657	47.9
	SVM	0.719	0.644	11.7
Precision	**DT**	0.948	0.800	18.5
	KNN	0.941	0.796	18.2
	LR	0.789	0.618	27.5
	NB	0.741	0.619	19.7
	NN	0.980	0.826	18.6
	SVM	0.756	0.607	24.6
Error Rate	**DT**	0.046	0.198	76.7
	KNN	0.209	0.397	47.4
	LR	0.253	0.395	35.9
	NB	0.366	0.388	5.82
	NN	0.024	0.180	86.9
	SVM	0.258	0.397	35.2

* **DT**: Decision Tree; **KNN**: k-Nearest Neighbors; **LR**: Logistic Regression; **NB**: Naive Bayes; **NN**: Neural Network; and **SVM**: Support Vector Machine.

Table 3.6: *Performance of the Detection Model on Thefts from New Users*

ML Model[a]	Accuracy	Recall (DR)	Precision	Error Rate
DT	0.883	0.885	0.780	0.117
KNN	0.715	0.660	0.763	0.285
LR	0.771	0.929	0.708	0.229
NB	0.628	0.396	0.667	0.372
NN	**0.929**	**0.932**	**0.999**	**0.071**
SVM	0.809	0.889	0.964	0.191

[a] **DT**: Decision Tree; **KNN**: k-Nearest Neighbors; **LR**: Logistic Regression; **NB**: Naive Bayes; **NN**: Neural Network; and **SVM**: Support Vector Machine.

achieves an excellent detection rate and minimal error rates with varying percentages of thieves. Our model shows an average detection rate of above 94% when 20% of the cluster users are thieves using decision trees and neural network classifiers.

3.5.5 Impact of Stealing Different Magnitudes of Electricity

In this final setting, we study how well our detection model behaves in cases where the stolen electricity amount ranges between low, medium and high levels. We consider that the stolen energy is low if it is below the 25% percentile of the overall consumed/generated energy. While we consider a stolen value to be high when it is above 75% percentile and the medium level is anything in between. In our experiments, "Low" is considered to be 300 and 100 watts in attacks #1, #3, #5, and #7 and to be 20% and 10% in attacks #2, #4, #6, and #8. In attacks #1, #3, #5, and #7, we choose the amount stolen to be between 500 and 900 watts in the case of "Medium" and to be between 1100 to 1500 watts in the case of "High". Whereas in attacks #2, #4, #6, and #8, we consider a percentage ranging between 30% to 50% and between 60% to 80% for "Medium" and "High" cases respectively. The results in Table 3.8 show that our detection model (using a neural network classifier) has an accuracy of 88.2% and a detection rate of 84.1% if the stolen electricity level is low. The model accuracy and detection rate increases to around 96% if the level of electricity theft is within the medium range. These results are typically expected as it would be easier for a detector to identify dramatic changes in the reported readings.

Table 3.7: *Performance of the Detection Model Under Different Percentages of Thieves*

Metric	Method*	Percentage of Thieves in a Cluster			
		2%	**5%**	**10%**	**20%**
Accuracy	**DT**	0.947	0.967	0.947	0.943
	KNN	0.943	0.937	0.941	0.917
	LR	0.800	0.811	0.755	0.781
	NB	0.829	0.732	0.604	0.672
	NN	**0.984**	**0.980**	**0.980**	**0.976**
	SVM	0.783	0.804	0.775	0.795
Recall (DR)	**DT**	**0.947**	0.935	1.000	0.927
	KNN	0.579	0.717	0.720	0.703
	LR	0.526	0.729	0.667	0.725
	NB	0.526	0.596	0.641	0.611
	NN	**0.947**	**0.936**	**0.936**	**0.957**
	SVM	0.684	0.681	0.638	0.694
Precision	**DT**	0.427	0.695	0.780	0.883
	KNN	0.122	0.266	0.427	0.635
	LR	0.085	0.194	0.310	0.453
	NB	0.032	0.100	0.135	0.376
	NN	**0.617**	**0.812**	**0.895**	**0.950**
	SVM	0.430	0.706	0.828	0.877
Error Rate	**DT**	0.027	0.023	0.029	0.028
	KNN	0.148	0.141	0.137	0.121
	LR	0.222	0.212	0.223	0.239
	NB	0.097	0.179	0.280	0.235
	NN	**0.013**	**0.012**	**0.012**	**0.011**
	SVM	0.027	0.022	0.025	0.066

* **DT**: Decision Tree; **KNN**: k-Nearest Neighbors; **LR**: Logistic Regression; **NB**: Naive Bayes; **NN**: Neural Network; and **SVM**: Support Vector Machine.

Table 3.8: *Performance of the Detection Model Under Different Theft Magnitudes*

Metric	Method*	Theft Magnitude Level		
		Low	**Medium**	**High**
Accuracy	**DT**	0.793	0.935	0.971
	KNN	0.628	0.806	0.900
	LR	0.612	0.729	0.757
	NB	0.584	0.633	0.700
	NN	**0.882**	**0.963**	**0.983**
	SVM	0.616	0.733	0.760
Recall (DR)	**DT**	0.798	0.932	0.967
	KNN	0.405	0.674	0.825
	LR	0.581	0.689	0.729
	NB	0.310	0.508	0.559
	NN	**0.841**	**0.958**	**0.971**
	SVM	0.597	0.700	0.715
Precision	**DT**	0.800	0.939	0.974
	KNN	0.706	0.917	0.970
	LR	0.627	0.760	0.769
	NB	0.670	0.695	0.790
	NN	**0.919**	**0.968**	**0.994**
	SVM	0.619	0.761	0.785
Error Rate	**DT**	0.207	0.065	0.029
	KNN	0.372	0.194	0.100
	LR	0.388	0.271	0.243
	NB	0.416	0.367	0.300
	NN	**0.118**	**0.037**	**0.017**
	SVM	0.384	0.267	0.240

* **DT**: Decision Tree; **KNN**: k-Nearest Neighbors; **LR**: Logistic Regression; **NB**: Naive Bayes; **NN**: Neural Network; and **SVM**: Support Vector Machine.

3.6 Threats to Validity

This section discusses the threats to the validity of the proposed approach. This means that we consider all factors or aspects in the research design that could potentially compromise the accuracy, reliability, or generalisability of the results. These threats could arise from various sources, such as limitations in the research design, biases in data collection, sample characteristics, measurement errors, or external influences.

The first threat concerns the creation of our dataset. The simulation may not fully capture the complexities and nuances of the real-world data, and thus the model's performance on actual data may differ significantly. To minimize this threat, we carefully selected our simulation tool and used a moderate-size taxonomy distribution feeder that outlines real-world characteristics. Another source of bias is the simulation of attacks chosen in our threat model. We had to focus on the attacks that are commonly used in the energy theft research area. Moreover, to avoid any bias in the generation of theft scenarios, we used the well-known mathematical model from [46]. We also randomly modified 50% of each user's data to reflect possible theft points. This can ensure the variability and generalisability of the attack scenarios as our dataset has almost 1600 users. Another potential bias arises in the selection of the theft magnitudes (values of l and k). In order to mitigate this, the experiment in Section 3.5.5 was added to reflect the validity of the proposed scheme when energy thieves steal different magnitudes of electricity.

While we acknowledge these threats to the validity of our study, we believe that we have sought plausibly to mitigate them and our results remain valuable and informative.

3.7 Summary

In this chapter, we proposed a data-driven energy theft detection mechanism in the presence of prosumers attacks. The detection approach is designed to detect different energy theft attacks from both consumers and prosumers by analysing reported SM readings in a cluster-based manner. Moreover, we introduced *balance attacks* which is a new attack scenario where attackers try to conceal their theft by balancing the total net of consumed or generated power. Simulations are carried out using a generated dataset comprising both prosumers' and consumers' generation and consumption profiles along with data from multiple data sources. Results show that the proposed model has a high detection performance for each type of attack and an overall 96%

detection rate. The detection model is also tested when different percentages of thieves are in a cluster. Results show that the proposed method achieves a good detection rate when the data tested is imbalanced. Although the obtained results support the *first hypothesis*, our proposed work here uses smart meter data without any considerations for *user privacy*. Fine-grained smart meter data may trigger serious privacy concerns, such as revealing users' presence/absence in their houses, or even their detailed daily habits at home. Therefore in the next chapter, we extend our proposed approach so that it can analyse data without compromising user privacy.

Chapter 4

Privacy-Aware Split Learning Based Energy Theft Detection

Effective detection of energy thefts is very important and must be implemented to comply with laws and regulations that govern users' privacy. Current detection approaches rely on significant amounts of raw fine-grained smart meter data and generally do not consider privacy. On the other hand, most privacy-preserving machine learning (PPML) approaches, such as homomorphic ML and federated learning (FL), are not well suited to the smart grid environment due to their processing complexity and communication overheads. Therefore, our contributions in this work are twofold: first, we propose a *privacy-preserving* detection model for energy thefts using the concept of split learning (SL). Subsequently, since classical split learning cannot be directly applied in the smart grid (SG) environment due to its communication overheads, we introduce a *new variant* of split learning, which we term as Three-Tier Split Learning (3TSL). This variant is more communication-efficient and suits the smart grid environment. The proposed model has two advantages over the existing techniques. First, the use of split learning enables the training of a detection model without the need to share raw data. This helps in achieving data privacy. Second, the splitting of the detection model allows the system to be more robust against honest-but-curious adversaries. Our evaluations show that the proposed detection model can ensure better privacy protection and communication efficiency, which are essential for smart grids, without compromising detection accuracy.

4.1 Introduction

Recently in the literature, several approaches have been proposed for the detection of energy thefts in the smart grid. However, as seen from our review in Section 2.7, very few have considered users' privacy in this regard (represented as F1 in Table 2.9), especially ML approaches [11, 34]. These approaches access users' raw energy data without any concerns for their privacy and ignore the fact that users' private data are governed by privacy policies such as GDPR. Using raw energy data creates new privacy vulnerabilities associated with what these data could reveal. For example, the disclosure of real-time, fine-grained power consumption can reveal the identity of an individual or information about his/her financial, social, physical or health characteristics [66]. In particular, high power consumption may reveal that people are in the house, while low readings may indicate that the house is empty. Hence, this creates the need to develop new mechanisms to be able to build energy theft detectors without violating users' privacy.

Moreover, our review revealed that research using privacy-preserving machine learning (PPML) in energy theft detection is still very limited and primarily relies on complicated cryptographic functions such as homomorphic encryption and MPC [11]. These methods are computationally and communicationally expensive and unsuited for smart meters, which are often computationally restricted [110]. Furthermore, the use of such cryptographic techniques alongside ML introduces additional processing and communication costs and would rely entirely on the strength of the key management mechanisms [59]. Moreover, distributed learning approaches to PPML, such as federated learning (FL) and split learning (SL), allow multiple parties to collaboratively train models without sharing their raw data. However, these techniques also have weaknesses. Recent research has shown that FL is prone to privacy attacks such as model extraction, membership-inference attacks, feature-inference attacks and label inference attacks [111]. This is especially true in an environment where aggregators and servers are considered honest-but-curious entities. An honest-but-curious entity is a type of adversary that is commonly used in the analysis of privacy properties. It is a legitimate participant of the system who will exactly follow the protocol defined but will attempt to learn all possible information from legitimately received communication [112]. Moreover, works that are based on using federated learning suffer from the following weaknesses:

- Communication is a critical bottleneck in the federated learning architecture. This is because it involves communicating a large amount of data between clients and the server in every round, where every message contains the complete model parameters.

- Each client in the federated learning approach needs to have a large storage capacity and high computational and communication capabilities to run a complete model. However, this is not the case in the smart meter environment, as devices will generally be resource-limited.

- Finally, privacy is often a major concern in federated learning. Sharing model updates (i.e. gradient information) instead of the raw data can reveal sensitive information to an eavesdropper or the system entities. This is a major problem faced in federated learning approaches, giving rise to what are typically referred to as "inference attacks".

Another limitation in the privacy-preserving energy theft detection research area is the lack of any quantitative analysis of that privacy (represented as F2 in Table 2.9). None appear to have addressed whether the privacy protection of an energy theft detector can be quantified. Whenever a privacy-preserving approach is used, such as encryption-based schemes and differential privacy, it is implicitly assumed that it provides strong privacy protection. On the other hand, differential privacy-based schemes suffer from a privacy-accuracy trade-off; and the privacy gained by these schemes is often proven using strict mathematical proofs. Whereas it has been shown that they are vulnerable to many privacy attacks [113, 114].

4.1.1 Our Contribution

In this work, we first propose an enhanced privacy-aware energy theft detection scheme which ensures users' privacy using the concept of split learning (SL). The classical SL approach has the advantage over FL in protecting users' data from reconstruction and feature inference attacks [62] since in SL, only the model updates of the split layer are sent rather than the whole model updated in the case of FL. SL is also less susceptible to model extraction and label inference attacks. However, SL cannot be directly applied to the environment of smart grids. This is because it introduces large communication overheads. Hence, we propose a new variant of SL called "Three-Tier Split Learning (3TSL)". In this variant, aggregators are intermediate entities in the system between clients and the central server. This architecture helps reduce the communication overhead of the system. It also makes the detection approach more suitable for smart grids where aggregators and energy suppliers are considered to be honest-but-curious entities. We also consider the issue of feature inference attacks in SL that has been studied in [115] and propose a defensive mechanism. We can summarise the contributions of this work as follows:

1. We propose an energy theft detection system which preserves the privacy of the users' data using split learning. The detection model combines stacked auto-encoders along with split learning to detect anomalies. This is the *first work* that applies split learning in energy theft detection.
2. We propose a *new variant* of split learning, called Three-Tier Split Learning (3TSL), that suits the nature of the smart grid infrastructure. This enhanced version adds aggregators to the system and splits the overall ML model into three parts (clients, aggregators, server) rather than two (clients, server). Moreover, we introduce a means of minimising the communication overhead by aggregating the updates from the split layers.
3. We evaluate our detection model with a range of different energy theft scenarios. This is also investigated in cases where malicious clients are involved in the training phase and a possible solution is discussed.
4. We analyse the privacy of the proposed model in terms of feature inference attacks and a metric called distance correlation.

In a nutshell, the major aim of our proposed scheme is to demonstrate how to achieve high-accuracy detection results while preserving privacy. The remainder of this chapter is organised as follows: some preliminary knowledge is defined in Section 4.2. We introduce the system and threat models in Section 4.3. In Section 4.4, we present our proposed privacy-preserving energy theft detection scheme. We detail our experimental setup in Section 4.5, while Section 4.6 gives the results. The threats to validity are discussed in Section 4.7, followed by a summary of this chapter in Section 4.8. All important notations used throughout this chapter are defined in Table 4.1.

4.2 Preliminaries

In this section, we briefly define distance correlation and indicate its usage. We used it as a quantitative evaluation metric for our privacy-preserving scheme. We also give some background information on the use of auto-encoders as anomaly detectors.

4.2.1 Distance Correlation

Distance correlation ($dCor$) is a statistical measure of dependence that was first introduced in [116]. This measure tests the joint independence between two random

Table 4.1: *Notations*

Symbol	Definition
CSM_i	The consumption smart meter reading of client i
PSM_i	The production smart meter reading of client i
W	The system's complete ML model function
d	Sample data point
$\bar{d}$	Reconstructed data point of d
$\hat{d}$	Modified sample data point (poisoned data)
y	Label of the sample data point d (benign or malicious)
o_i	Outputs (activations) of client i
o_a	Outputs (activations) of the aggregator
W^{-1}	An attacker's inference model

vectors X and Y with arbitrary dimensions (lengths). Distance correlation captures linear and nonlinear relationships between the variables, making it a good measure of independence in our case. Specifically, we aim to quantify the dependency between the split layer outputs and the original readings. Unlike the classical definition of correlation, in distance correlation, we get a value between 0 and 1 where zero indicates total independence between the two vectors. Distance correlation can be calculated by dividing the distance covariance of the two variables by the product of their distance standard deviations. This makes the distance correlation between two random variables X and Y equals to:

$$dCor(X,Y) = dCov^2(X,Y)/\sqrt{dVar(X)\,dVar(Y)} \tag{4.1}$$

where $dCov(X,Y)$ is the distance covariance between X and Y and $dVar(X)$ is the distance covariance between X and itself, i.e. $dCov(X,X)$. The distance covariance $dCov(X,Y)$ is the square root of the average of the product of the double-centred pairwise Euclidean distance matrices and can be calculated as:

$$dCov^2(X,Y) := \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} D(x_i,x_j)D(y_i,y_j) \tag{4.2}$$

where the $D(x_i,x_j)$ is the "centred" Euclidean distance between the ith and jth observations and can be calculated as:

$$D(x_i,x_j) = ||x_i - x_j|| - \bar{a}_{i.} - \bar{a}_{.j} + \bar{a}_{..} \tag{4.3}$$

where $\bar{a}_{i.}$ is the ith row mean of the distance matrix of X, $\bar{a}_{.j}$ is the jth column mean of the distance matrix of X and $\bar{a}_{..}$ is its grand mean.

4.2.2 Anomaly Detection Using Auto-Encoders

An auto-encoder (AE) is a special type of neural network that has mainly two parts, an encoder part and a decoder part. (Refer to Appendix A for more details about how a neural network works.) The encoder compresses the input features to produce a latent representation that is then decoded by the decoder to reconstruct the input features [117]. Auto-encoders are trained to minimise the difference between the reconstructed data and the original input where the model learns the relationships among features of the input set. This difference between the original input d and the reconstructed data $\bar{d}$ is called the reconstruction error (RE). It can be measured using any measurement of error such as the absolute error $RE = ||d-\bar{d}||$ or the squared error $RE = ||d - \bar{d}||^2$ [118]. After the model has converged, the reconstruction error can be used to detect anomalies. Auto-encoders are best suited for anomaly detection in environments with high-volume data streams such as smart grids. They can be trained to learn the representation of a single 'normal' class. Attacks (or at least anomalies) can be detected without labelling by observing the magnitude of the reconstruction error [119]. Reconstruction errors for normal data points are minimised by the auto-encoder whereas anomaly-input data result in higher reconstruction errors. A suitable threshold is required to assess if the errors are high enough for that data to be termed anomalous [119]. Stacked autoencoders (SAEs) are constructed by stacking several AEs together. The first AE maps the input to a first latent representation. After training the first autoencoder, its decoder layer is discarded and then replaced by a second autoencoder, which has a smaller latent vector dimension. This process is repeated depending on the depth of the SAE. The depth of stacked autoencoders helps in learning more abstract features from the extracted ones [117].

4.3 System Model and Threat Model

4.3.1 System Model

As we mentioned in Chapter 3, we consider a classical smart grid model with three entities: clients, substation gateways (aggregators) and the control centre (server). Here, the gateways act as aggregators which are responsible for processing the clients' data before sending them to the server Moreover, the data sent between the three entities (smart meters, aggregators and server) are in the form of model updates, i.e. activations and gradients rather than raw SM readings since we are employing a variant of split learning to our system model. Further details are explained in the proposed theft detection model section.

4.3.2 Threat Model

In this work, we consider a practical threat model with multiple security and privacy attacks where all external and internal entities can act maliciously. The system assumes that aggregators and the server are honest but curious entities (semi-honest), i.e. they do not tamper with the system's instructions but they may try to infer information about users' behaviours. External adversaries may eavesdrop on the activations sent between the system's entities in an attempt to learn individuals' private data. We also assume that participating clients have the ability to modify and manipulate their smart meter readings or their neighbours' readings in an attempt to gain financial advantage. These attacks can be viewed as follows:

- *Energy Theft Attacks*: These attacks occur when a user (either a consumer or a prosumer) tries to modify the reported smart meter's readings. We consider the same set of attack scenarios explained in our previous work in Chapter 3. In total, we consider eight different attack types of energy thefts.
- *Poisoning Attacks*: These can be carried out only by internal clients that can modify the smart meter data. The goal of this attack is to compromise the performance of the detection model W and to cause it to make incorrect predictions by using crafted data points $\hat{d}$. Let W be an ML function which maps D to Y, $W(d) = y$ where y is the correct label and d is a real unpoisoned data point. When the model W is trained using a set of poisoned data points $\hat{D}$, W diverts from its normal behaviour and produces wrong outputs $\hat{Y}$. In practice, energy theft attacks discussed before (including balance attacks) are types of poisoning attacks.
- *Feature Inference Attack*: This attack compromises the privacy of the users' readings and can be launched by external or internal adversaries. In our system, we consider the server to be a trusted organisation (e.g. the National Grid in the United Kingdom), whereas most clients and all aggregators are honest-but-curious parties. We call these honest-but-curious entities as *passive adversaries* since they correctly follow the steps of the proposed model but try to passively perform *feature inference attacks* on data outputs that other clients send. The goal of the attack is to guess the values of the sensitive features of a data point given only the activations sent by the client (i.e. its model component's split layer activations). A formal definition of the attack in the context of split learning is as follows:

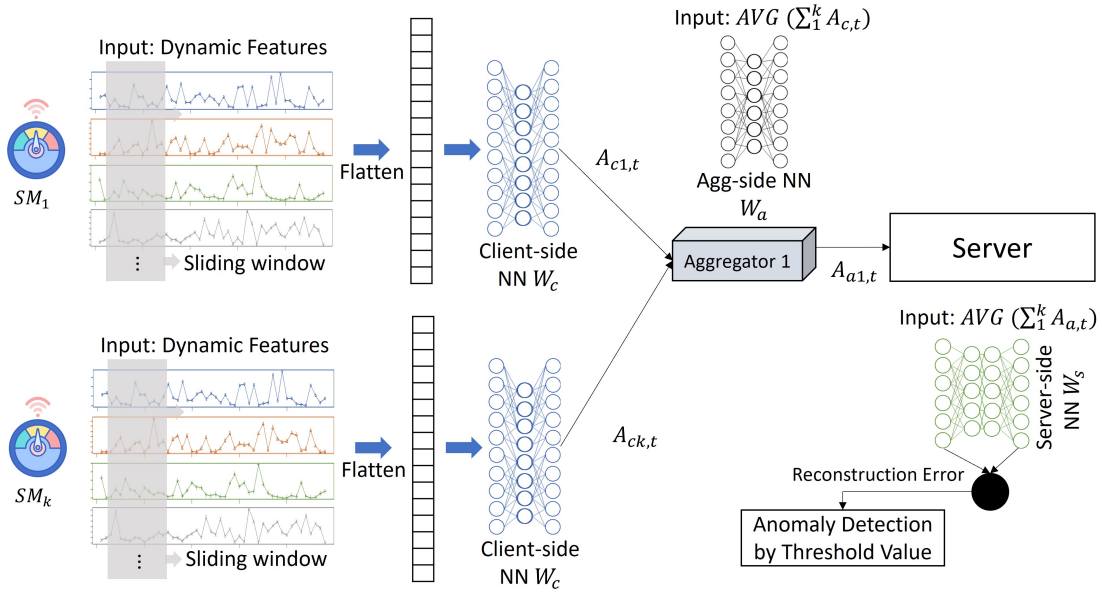


Figure 4.1: *Proposed Theft Detection Model*

Definition (Feature Inference): Let d be an input data point with a set of features $d_1, d_2, ..d_n$ where $d = (d_1, d_2, ..d_n) \in D$, and let W_c be a client model that maps D to O, $W(d_i) = o_i$ where d_i is the input data of client i and o_i is this client's activations or outputs from the split layer. To launch a feature inference attack, the attacker tries to find a function W^{-1} that can infer d_i from o_i, $W^{-1}(o_i) = d_i$. The goal is to have an exact inference. However, in practice, useful inference can be approximate.

Feature Inference Attacks (FIAs) have caught the attention of privacy and security researchers after the widespread adoption of collaborative machine learning models in different applications. It has been studied in [120, 121, 122, 123] and by many others in the machine learning community. However, it has not been studied or considered in the area of privacy-preserving machine learning energy theft detection.

4.4 Proposed Theft Detection Model

In this section, we explain the "Three-Tier Split Learning (3TSPL)" approach, which is the newly proposed variant of split learning and describe the theft detection model. Figure 4.1 shows how the proposed detection model works.

4.4.1 Three-Tier Split Learning (3TSL)

Our Three-Tier Split Learning architecture follows the system design of the state-of-the-art split learning system but adds one new component which is an aggregator between the clients and the server. The newly added aggregator makes the split learning framework more applicable to the context of smart grids. We also introduce a way to calculate the intermediate updates by averaging the activations received from the clients for each client-aggregator pair before sending the results to the server. This makes the process more parallel than sequential. In our extension of split learning, the learning model W is split into 3 different parts, W_c at the client side, W_a at the aggregator and W_s at the server side. The procedure of the 3TSL method starts as follows: each client c trains the W_c part of the network and sends the activations of the split layer to the aggregator a. Each aggregator a waits until it receives all activations from its clients and computes the average of these activations and uses it to complete a forward pass on its part of the model W_a. After completing a forward pass, the aggregator sends the activations of the last layer of its model to the server. As in the aggregator, the server waits for the activations from all aggregators and computes their average to be used as input for its part of the model W_s. After the completion of the forward pass, the server generates the gradients for the final layer and back-propagates the error to its cut layer of W_S. The gradients are then passed to the aggregators who perform a back-propagation and send their gradients to the clients. The rest of the back-propagation is completed by the clients. This process is continued until the model converges. Algorithm 1 provides detailed instructions for the "Three-Tier Split Learning".

4.4.2 Energy Theft Detection Approach

The aim of this work is to explore how split learning can be used to train an anomaly detector to detect energy thefts without violating clients' privacy. We do so by combining the previously explained 3TSL method with a stacked auto-encoder. The stacked auto-encoder (SAE), as an unsupervised ML algorithm, enables us to train the detection model without the need for data labels, and the 3TSL provides privacy assurance as clients will not need to send their private raw data.

The architecture of our energy theft detection model is shown in Figure 4.1. A stacked auto-encoder (SAE) model is split between the system's entities, one part is at the client's side, the second part is at the aggregator's side and the third part is at the server's side. The server part consists of 4 layers, while we use varying depths for the client's and aggregator's parts (details can be found in Table 4.3). Next, each client collects a set of features that includes consumption, generation and weather

Algorithm 1 Three-Tier Split Learning (3TSL) Algorithm with Averaging

function SERVER ▷ executes at round $t \geq 0$
  for epoch e **do**
    $A_t \leftarrow []$
    for agg $a \in aggregator_t$ **do**
      $A_{a,t} \leftarrow \text{AGGREGATOR}(a, t)$
      $A_t[c] \leftarrow A_{c,t}$
    end for
    $A_{t.avg} \leftarrow sum(A_t)/len(S_t)$
    Complete forward propagation with $A_{t.avg}$ to get $A_{S,t}$
    Calculate Loss
    $W_{S,t+1} \leftarrow W_{S,t} - \eta\nabla\ell(W_{S,t}; A_{t.avg})$ ▷ Back propagation part of the server
    $\text{CLIENTBACKPROP}(c, t, \nabla\ell(A_{t.avg}; W_{S,t}))$ ▷ k here is the last client
  end for
end function

function AGGREGATOR(a,t) ▷ executes at round $t \geq 0$
  for epoch e **do**
    $A_t \leftarrow []$
    for client $c \in S_t$ **do**
      $A_{c,t} \leftarrow \text{CLIENTUPDATE}(c, t)$
      $A_t[c] \leftarrow A_{c,t}$
    end for
    $A_{t.avg} \leftarrow sum(A_t)/len(S_t)$
    Complete forward propagation with $A_{t.avg}$ to get $A_{a,t}$
    send $A_{a,t}$ to Server
  end for
end function

data at regular intervals (usually every 15-20) minutes. And given the nature of time-series data, a sliding window of multiple data points is considered as an input to the client's part of the model. This helps capture the correlation between consecutive data points. After that, the vector of features is fed to the client's part of the SAE and the latent representation (client's output) is sent to the aggregator. Each aggregator uses the average of all the outputs of its clients as an input to its part of the SAE. The output of the aggregator, which is the second latent representation, is sent to the server, which also uses the average of all the aggregators' outputs as the input to its part of the SAE. This process is repeated until the model converges.

To detect energy thefts, the server computes a threshold which is used as a bound to detect those energy thefts. Any data point that causes a reconstruction error exceeding this predefined threshold would be considered as an anomaly. Here, mean squared error (MSE) is used as the reconstruction error function. To estimate the detection threshold, we use the same procedure as in [117] where the server calculates the reconstruction errors of its part of the SAE for the whole training dataset. Then, the threshold value is estimated by the mean and standard deviation of those reconstruction errors; it can be described as:

$$threshold = \frac{1}{x}\sum_{i=1}^{x} RE_i + \sqrt{\frac{1}{x}\sum_{i=1}^{x}(RE_i - \frac{1}{x}\sum_{i=1}^{x} RE_i)^2} \tag{4.4}$$

where RE is the reconstruction error of the server's part of the model (the difference between the server's input and output), and x is the number of training elements that the server uses.

4.5 Experimental Setup

In this section, we give details about how we conducted our experiments, such as the dataset used, the formation of the energy theft attacks, the simulation environment, neural network parameters, and evaluation metrics.

4.5.1 Dataset

In this work, we have reused the dataset from Chapter 3. The dataset includes the energy profiles of 1596 clients, 49 of whom are solar panel prosumers. Every client reported 17 different dynamic parameters and 13 physical parameters of each client's property every 15 minutes. In this work, we used a sliding window of 16 data points (4 hours) as input to the client's part of the model. This means that each sample is a vector of 285 features ([17 dynamic features * 16 data points] + 13 static features). These data samples are split into 70% for training and 30% for testing. The rationale behind employing a 70-30 train/test ratio instead of 80-20 is primarily due to the use of the sliding window concept. This greatly increased the number of features in each data point, from 16, as seen in Chapters 3 and 5, to 285. By incorporating time series windowing, the quantity of data points available for evaluation was reduced. Therefore, a larger portion (30% as opposed to 20%) of the dataset had to be allocated for performance testing purposes.

4.5.2 Energy Theft Attacks

Since all readings in our dataset are unmanipulated (normal) points, they were modified using the same approach described in Section 3.4.2. In this work, the value of l is chosen to be 400 which is approximately one-third ($\frac{1}{3}$) of the mean of all readings. While the value of k is set to 40 which is less than half ($\frac{1}{2}$) of the reading.

4.5.3 Simulation Environment

The proposed 3TSL detection model is implemented using PyTorch [124]. PyTorch is a Python-based machine learning library that enables access to every computational node in an ML model. This allowed us to split the whole detection model into three splits.

4.5.4 Neural Network Parameters

In our experiments, the SAE model consists of a total of ten neural network layers. In our approach, clients had three layers, aggregators had three layers, and the server had four layers; however, varying neural network depths were used for the client and aggregators in the experiment in Section 4.6.3. In every experiment, the mean squared error (MSE) is used as the model loss function. The training phase iterated over the samples for a total of 20 epochs with a batch size of 96. The Adam optimiser [125] is used with its default hyper-parameters as the model optimization algorithm in all components.

4.5.5 Evaluation Metrics

To evaluate the performance of the proposed model in terms of energy theft detection, we consider accuracy, recall (also known as detection rate (DR)), and precision. These basic metrics allow the calculation of other metrics using them, such as F1 or F2 scores [126]. We also used distance correlation as a metric for evaluating the privacy preservation level gained by our proposed model. These are the metrics that we report. However, we used MSE as our RE and loss function to derive these results.

4.6 Results and Discussion

In this section, we analyse the security and privacy of our proposed model against the threat model and provide computational and communication analysis.

4.6.1 Detection of Energy Thefts Attacks

Here, we evaluate how well the SAE works in our "Three-Tier Split Learning" setting in terms of energy theft detection. For this purpose, we have also trained the same SAE in two other settings: a centralised setting and a federated learning setting. In the centralised setting, the global model of the SAE along with all client data is available at the server side. This setting is the basic setting where privacy is not considered. In the federated learning setting, the SAE model is trained locally at each client and a shared model is averaged at the server side.

Figure 4.2 compares the performance of our proposed model with the centralised version and the federated learning approach in terms of accuracy, recall and precision. It is clear from the figure that the results of our proposed approach are highly comparable to the other two settings. This shows clearly that our approach achieves excellent results in detecting energy thefts while preserving privacy compared to the centralised approach. The results are also very similar to the federated learning approach with the advantage of having lower communication overhead (discussed in Section 4.6.7). In all three settings, the training is taking place repeatedly over different batches of data and therefore the results are not linearly improving.

4.6.2 Resilience Against Poisoning Attacks

As explained in Section 4.3.2, poisoning attacks are attacks that can be launched whenever a collaborative ML algorithm is involved. In our approach, these attacks are possible because clients are involved in training the detection model. In this experiment, we have tested how well our detection approach works in the event of having poisoned training data. We tested this using different percentages of poisoned training data ranging from 0% (which means that all clients are honest and there is no poisoning attack) to 20% (which means that each client poisons 20% of their data). Our results in Table 4.2 show that the more poisoned data is used to train the system, the worse our detection results are. When only 20% of the data is poisoned, then the detection rate decreases by almost 15%. Therefore, we had to find a way to overcome this. We adopt a simple solution where we randomly drop 10% of the training updates

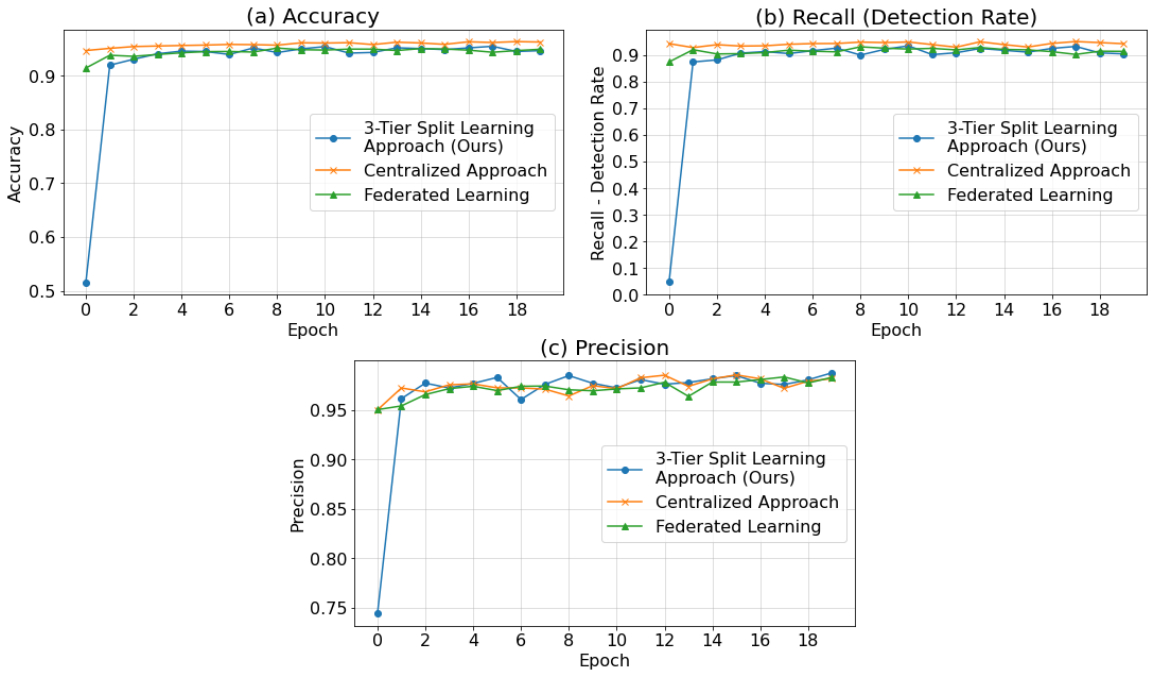


Figure 4.2: *Results of the Detection Model Using Three-Tier Split Learning (Proposed Work), Centralised Detection, and Federated Learning*

received from the clients. As can be seen in the last record of Table 4.2, this simple random dropping improves the detection results and make it comparably close to the normal case where no poisoned data are injected.

4.6.3 Privacy Analysis via Distance Correlation and Feature Inference Attack

In order to analyse the privacy aspect of our model, we explored the feature inference attack where we analyse the correlation between the activations sent in the system and how much they leak the original raw data. Several metrics can be used to quantify the correlation between variables, such as Pearson's correlation, Spearman's rank correlation, distance correlation and Phi coefficient. In this context, we use *distance correlation* as explained in Section 4.2.1. This is because distance correlation is one of the few statistical measures that can test the dependence of two arbitrary length vectors (e.g. raw data and the split layer's activations). It can also show both linear and nonlinear associations, which makes our evaluation more comprehensive. Distance correlation was used in [127] and [128] as part of their privacy assessment frameworks. It takes a value between 0 and 1, where lower values indicate greater

Table 4.2: *Detection Results with Poisoned Data*

Percentage of Poisoned Training Data	Accuracy	Recall (DR)	Precision
0% (No attack)	**0.946**	**0.905**	**0.970**
5%	0.945	0.901	0.969
10%	0.933	0.891	0.973
15%	0.870	0.760	0.974
20%	0.870	0.762	0.973
20% with random 10% dropping	**0.933**	**0.903**	**0.979**

independence of the two vectors. Our goal here is to make the distance correlation value less than 0.5.

Table 4.3 shows the distance correlation $dCor$ between the outputs (activations) sent by the client o_i and the real raw input data d_i. It also shows the $dCor$ between the outputs sent by the aggregator o_a and the raw input data. In the first setting, both the client and the aggregator have 3 hidden layers with no dropout layers in between and as the results show, the distance correlation between the raw data and the output sent by the clients is high (0.74). This actually suggests that it would be easy for the attacker to infer the raw data back from those activations. One possible defence was to employ dropout [129], and another one was to increase the number of hidden layers. In neural networks, Dropout is a well-known regularisation technique that is used to overcome overfitting [130]. The basic idea of dropout is to randomly deactivate neurons' activations with a probability between 0 to 1. This random dropout of activations will make it harder for the attacker to build a robust system that can infer the raw data from the activations as the attacker will be observing a different activations list each time [129]. As you can see in Table 4.3, after adding some dropout layers and increasing the number of hidden layers the distance correlation is decreasing and at the same time this does not affect the theft detection rate (recall) in any way. Briefly, we can say that our 3TSL approach can protect against feature inference attacks, defined in Section 4.3.2.

Protection Against Feature Inference Attacks: Suppose an adversary obtains the set of outputs sent from the client to the aggregator o_i. He/She needs a function W^{-1} that can infer the original raw data d_i from o_i, $W^{-1}(o_i) = d_i$. However, our results show that the average $dCor(d_i, o_i)$ is less than 0.5 with only 4 layers and

Table 4.3: *Feature Inference Analysis*

Model Used	***dCor(*d_i*, o_i)***	***dCor(*d_i*, o_a)***	**Recall (DR)**
Client: 3, Agg: 3, No Dropout	0.740	0.369	0.937
Client: 3, Agg: 3, 1 Dropout	0.620	0.359	0.945
Client: 3, Agg: 3, 2 Dropout	0.557	0.366	0.935
Client: 4, Agg: 4, No Dropout	0.665	0.342	0.937
Client: 4, Agg: 4, 2 Dropout	**0.490**	**0.340**	**0.934**

dropout at the client model. This implies that the probability of finding W^{-1} with good accuracy for any probabilistic polynomial time adversary $\mathcal{A}$ is negligible, i.e., $Adv_{\mathcal{A}}^{leak}(d_i, o_i) \leq \epsilon$.

4.6.4 Analysis of Detecting Different Magnitudes of Electricity Theft

In this set of experiments, we tested the performance of our energy theft detector in detecting different magnitudes of stolen energy. The results are shown in Table 4.4. In these experiments, we tested our model with different values of stolen electricity l and k such that l ranged between 700 to 100 and k ranged between 0.7 and 0.1, where $l = 700$ and $k = 0.7$ indicates a high volume of stolen electricity and when $l = 100$ and $k = 0.1$, it indicates a very low volume of stolen electricity. The results demonstrate that our detector is able to detect energy thefts even in low volumes (as low as 200 Watts). However, it fails to detect lower volumes of energy thefts (100 Watts). There will always be a threshold below which thefts cannot be detected reliably without significant false positives. Stealing 100 watts hours every 15 minutes will cost only around 20 pence per hour (according to current UK energy prices in 2023) [131] and it would take a thief around seven months (5,000 hours) to steal £1000. Though this is not a trivial sum, in practice thieves are decidedly constrained.

Table 4.4: *Analysis of the Detection of Different Intensities of Energy Thefts*

Theft Intensity	Accuracy	Precision	Recall	F1 Score
l=700 AND k=0.7	0.96	0.92	1.00	0.96
l=600 AND k=0.6	0.95	0.92	0.99	0.96
l=500 AND k=0.5	0.95	0.92	0.99	0.95
l=400 AND k=0.4	0.95	0.92	0.98	0.95
l=300 AND k=0.3	0.93	0.92	0.94	0.93
l=200 AND k=0.2	0.83	0.90	0.74	0.81
l=100 AND k=0.1	0.57	0.73	0.23	0.35

4.6.5 Trade-off Between Privacy and Detection Accuracy

We experiment with the trade-off between minimising the distance correlation (improving privacy level) and maximising the detector's accuracy. In this experiment, we modified the objective function of our ML model to be:

$$min(MSE + \alpha * dCor) \tag{4.5}$$

where the value of α corresponds to the degree of privacy that we want to achieve.

Table 4.5 shows the results when we set α to different levels ranging from 0 to 1. When alpha is set to 0, it means that the model is trained without any consideration for privacy, while if it is set to 1, it means that we are trying to minimise the *dCor* level between the client's output and the original inputs as much as possible. As can be seen from the results in the table, there is a clear trade-off between privacy and energy theft detection performance. The higher we aim for privacy (by minimising *dCor*) the lower the accuracy of our detection model. After experimenting with different values of α ranging between 0 to 1, we can see that a good balance is given when α is set to 0.01. It gives excellent energy theft detection performance and an excellent reduction of the *dCor* level compared to the original results (where $\alpha = 0$).

4.6.6 Computational Overhead

One main objective of applying split learning instead of federated learning is to ensure that the proposed model does not introduce much computational overhead on the

Table 4.5: *Analysis of the Trade-off Between Detection and Privacy*

α	Accuracy	Precision	Recall	F1 Score	AVG $dCor$
0	0.93	0.90	0.96	0.93	0.893
0.01	0.91	0.90	0.93	0.92	0.234
0.05	0.79	0.83	0.70	0.70	0.233
0.1	0.66	0.79	0.44	0.57	0.146
1	0.49	0.46	0.09	0.15	$1e^{-7}$

resource-constrained smart meters. In this experiment, we investigated the proposed model overhead in terms of computation time and compared it with the federated learning approach (as it is the closest scheme to ours in providing security and privacy properties). We computed the computational time of a single round carried out in a smart meter. The experiments were repeated multiple times to ensure statistical significance, and the average execution time for each model was presented.

It can be seen from Table 4.6 that the computational times for computing a single round of 3TSL and federated learning in a smart meter are 0.0029 and 0.0163 seconds, respectively. This means that the federated learning scheme executes five times slower than our proposed one. This can be attributed to the fact that in federated learning, the smart meter is responsible for computing the whole model instead of only part of it as in the case of three-tier split learning. Therefore, the results here confirm the advantage of using split learning in lowering the computational overhead at the client side.

Table 4.6: *Computational Overhead*

Method	Computational Time (in seconds)
Three-Tier Split Learning (Ours)	0.0029
Federated Learning	0.0163

4.6.7 Communication Analysis

The objective of the proposed 3TSL theft detector is not only to ensure privacy and non-thefts by the smart meters but also to ensure that the communication overhead is minimised. In this section, we compare the communication overhead of the proposed scheme with the classical split learning approach (where aggregators are assumed to only forward clients' communications to the server without any averaging) and with the federated learning approach (as it is the closest scheme for providing security and privacy properties to the system). To do so, we analyse the amount of data transferred by every client and the total data transferred between parties in the system. We use the following notation to mathematically measure communication efficiency. Notation: K= # clients, L= # aggregators, N= size of the complete model parameters (neurons), S_c= size of the split layer at the client, and S_a= size of the split layer at the aggregator. Here $K > L$ and $N \gg S_c + S_a$.

In Table 4.7, we can see that the communication cost, for the same neural network model, in the three-tier split learning approach is less than that of both the classical split learning and the federated learning approach. In the 3TSL, every client sends the updated activations from their split layer S_c and receives the updated gradients from the aggregator with size S_c, which totals $2S_c$. The same number of communication applies for every aggregator which makes the total communication of one round equal to $K \times (2S_c) + L \times (2S_a)$. In the classical split learning approach, when averaging is not implemented, the aggregator would act as a repeater, forwarding every communication between the server and the client. This will make the clients' updates and the gradients' updates be sent twice in the network making the total communication in classical split learning greater than our proposed three-tier split learning. By aggregating the model updates at the aggregator side in the proposed 3TSL scheme, we are reducing the communication to the server. Thus protecting it from various attacks such as denial of service. On the other hand, clients in the federated learning approach send the full network updates to the server and the full gradients are then forwarded from the server to all clients. This makes the total communication of one round equal to $2KN$ which is significantly more than $K \times (2S_c) + L \times (2S_a)$.

4.6.8 Summary of Comparison

In summary, our results show that the proposed approach outperforms existing ones in terms of detection, privacy and communication overhead. This comparison is outlined in Table 4.8. Our 3TSL approach can detect energy thefts with high recall, precision, and accuracy. It also preserves the privacy of users' data as compared to the

Table 4.7: *Communication Analysis*

Method	Communication Per Client	Total Communication
Three-Tier Split Learning (Ours)	$2S_c$	$K \times (2S_c) + L \times (2S_a)$
Classical Split Learning	$2S_c$	$K \times (2S_c) + K \times (2S_c)$
Federated Learning	$2N$	$2KN$

centralised approach. In terms of resilience against poisoning attacks, all three models can detect poisoning attacks with the help of an additional procedure (for instance, in our case by adding dropout layers and randomly dropping 10% of the training data). Moreover, it is more challenging to infer features from only the split layer's activations than from the whole model updates [111]; hence our proposed model provides stronger resilience against feature inference attacks than the federated learning approach.

Furthermore, our computational and communication analysis (provided in Sections 4.6.7 and 4.6.6) shows that the proposed approach has lower computational overhead and higher communication efficiency than the federated approach. It should be noted that the communication efficiency of our model would also be better than the centralised approach when the feature set is larger than the split layer size, which is true in most cases.

4.7 Threats to Validity

In this section, we discuss the possible validity threats for our developed work. The first thing to note is that the threats to the validity of our proposed work outlined in Chapter 3 still hold here and their mitigation measures were also applied in this work. A new threat to validity arises here from the choice of privacy evaluation measurement. This choice may fail to accurately capture the intended concepts. Quantifying privacy poses challenges, as it may not be applicable to all types of data or systems. In our work, we chose a metric that is objective and can be applied to different domains.

Table 4.8: *Comparison*

Property	Centralised Approach	Federated Learning Approach	Three-Tier Split Learning (Ours)
Energy theft detection	✓	✓	✓
Privacy preservation	✗	✓	✓
Resilience against poisoning attacks	✓	✓	✓
Stronger resilience against feature inference attacks	✗	✗	✓
Lower computational overhead	-	✗	✓
Higher communication efficiency	✓✗	✗	✓

✓: True. ✗: False. ✓✗: This may not be true in some cases.

4.8 Summary

This work proposed an approach that supports our *second hypothesis*. We have presented a new variant of split learning, Three-Tier Split Learning, as a private collaborative machine learning algorithm to tackle the challenge of preserving users' privacy, where it trains a detection model for energy thefts without the need to share raw data. It is tested on a dataset that contains malicious readings generated from various cyber-attacks, including consumer thefts, prosumer theft and balance attacks. Our experiments showed that it gives a 94.6% accuracy, 90.5% recall (detection rate) and 97.0% precision. Moreover, even in the case of poisoning attacks, simply dropping 10% of the model updates can provide comparable results to those with no poisoned data. The model demonstrates good privacy preservation where the distance correlation between the updates sent from the clients/the aggregator and the raw data is low, making it difficult for attackers to infer the raw data from those updates. There is also a significant reduction in the number of messages sent and received to/from the server. Thus, our proposed model ensures privacy preservation and communication efficiency. When both privacy and energy theft detection are achieved, effort is needed for post-detection procedures that ensure reliable management of other grid operations. Our next chapter of this thesis will propose a combined energy theft detection and demand management scheme with stronger privacy levels than what is achieved here.

Chapter 5

Privacy-Enhanced Energy Theft Detection for Effective Demand Management

The detection of energy thefts is vital for the safety of the whole smart grid system. However, the detection alone is not enough since energy thefts can crucially affect the electricity supply leading to some blackouts. Moreover, as pointed out in the previous chapter, privacy is one of the major challenges when dealing with clients' energy data. However, it is often overlooked as most current detection techniques rely on raw, unencrypted data, which may potentially expose sensitive and personal data. In Chapter 4, we explored the use of an enhanced version of split learning to detect energy theft. Here, we extend the idea and present a more privacy-preserving energy theft detection technique with *effective demand management*. To make our model more privacy-preserving, we employ a second layer of privacy that masks clients' outputs to prevent inference attacks. Another *privacy-enhanced version* of this mechanism is also proposed here. It provides an additional layer of privacy protection by training a randomisation layer at the end of the client-side model. This makes the output as random as possible without compromising the detection performance. For the energy theft detection part, we design a multi-output machine learning model to detect energy thefts, estimate their volume, and effectively predict future demand. Finally, we use a comprehensive set of experiments to test our proposed scheme. The results show that our scheme achieves high detection accuracy and greatly improves privacy preservation.

5.1 Introduction

Energy thefts have non-financial consequences. For instance, they were the cause of 15 different blackout incidents in the US in 2017 alone [132]. Moreover, energy thefts can severely affect demand management, leading to mis-forecasting of future electricity supply. As a result of underestimating the supply, some grid regions will experience blackouts. Overestimating the supply may give rise to extreme and typically infeasible storage requirements. Such storage (typically via batteries) may also be very expensive. Despite these consequences, solutions to the energy theft problem generally focus solely on detection measures.

Over recent decades, several works have been proposed in the literature on demand-response management in smart grids, such as [133, 134, 135]. Authors in [110, 136] had also considered privacy-preserving approaches for their solutions. However, all of these solutions assumed that the data supplied by the clients were genuine. None has considered when energy thefts are present in the system and to what degree these thefts would impact demand-supply management. Similarly, most proposed energy theft detection techniques do not consider privacy and use energy usage data in its raw form. However, since such use of data can lead to privacy risks, there was a need to address privacy in energy theft detection research. A few papers have addressed privacy-preserving ML-based (PPML) energy theft detection in SG; these have been discussed in Section 2.7.2. We divided them into two broad categories: the first use cryptographic-based methods, while the others are based on distributed machine learning techniques.

Investigating the existing literature on both demand management and energy theft detection research areas reveals two major issues: (i) Although several works have been proposed in demand management for smart grids, they have yet to consider the issue of managing the demand in cases where energy thefts exist. Instead, the existing research has always assumed that all clients are honest and would report reliable readings, which is not always true. (ii) On the other hand, although several solutions have been developed for energy theft detection, all the research in this area has focused solely on the detection part and has not gone beyond that. However, it is essential to act upon detecting energy theft, including estimating the amount of stolen energy and considering it while forecasting the future energy demand. In fact, Ofgem[1] had set some rules for tackling electricity theft, one of which requires all energy suppliers in the UK to *"make accurate estimates of the volume of electricity stolen following detection"* [137]. This is an essential post-detection step that no one

[1]Ofgem is the Office of Gas and Electricity Markets and it is the energy regulator for Great Britain.

has considered before, as indicated in our literature survey in Table 2.9. To the best of our knowledge, we are the first to propose such a step.

Moreover, solutions that aim to achieve privacy in energy theft detection using cryptographic methods or federated learning have their weaknesses that were highlighted previously in Section 4.1.

5.1.1 Our Contribution

To address all the limitations mentioned above, we now propose a privacy-preserving energy theft detection model that can effectively predict energy demand. Our proposed model not only detects energy thefts of different types but also helps to reliably manage the power demand even in the event of thefts. This is the first work to develop a solution that bridges the gap between the energy theft detection and demand management research areas. In this work, we first propose an energy theft detection system that preserves users' privacy by using split learning as the architecture of our machine learning approach. This approach detects energy thefts, estimates the amount of stolen energy and manages the future demand even in cases of thefts. Moreover, to avoid the issue of feature leakage, we utilise a lightweight masking approach to lower the chances of any inference attacks. An enhanced privacy-preserving approach is also proposed by using an added neural network layer that is trained to randomise the outputs of the client's part of the model. In both proposed designs, users' data are kept private while the system can still detect energy thefts.

The remainder of this chapter is organised as follows. In Section 5.3, we explain the details of the system and threat models employed in this work. In Section 5.4, we give a detailed description of the proposed privacy-preserving energy theft detector with the demand-management model. In Sections 5.5 and 5.6, we present our experimental setup and results indicating both the detection abilities of the proposed approach and its privacy properties. We discuss threats to validity in Section 5.7. Finally, we conclude the chapter in Section 5.8.

5.2 Preliminaries

In this section, we provide an overview of the pseudorandom number generator (PRNG) algorithm and the quantisation process to be used in the proposed model.

5.2.1 Pseudorandom Number Generation

A Pseudorandom Number Generator (PRNG) algorithm produces a sequence of numbers that appear to be random but are generated using a deterministic process. Unlike true random number generators, PRNGs start with an initial value and use mathematical operations to produce a sequence of numbers that, while not truly random, exhibit properties of randomness. Linear congruential generators (LCGs) are commonly used PRNG algorithms with sequences generated using the formula $X_{n+1} = (A(X_n) + C)\ mod\ M$, where X_n is the current number, A is a multiplier, C is an increment, and M is the modulus. The choice of parameters significantly affects the quality of the generated sequence.

5.2.2 Quantisation

Quantisation is a process used to represent continuous values (floats) with discrete, quantised values. It involves dividing the range of continuous values into a finite set of discrete levels or intervals. Specifically, it is done by mapping the min/max of the continuous values with a chosen min/max threshold of the integer range $[-\beta, \beta]$. This transformation introduces a trade-off between accuracy and data storage or transmission efficiency, where higher precision, or bigger β numbers, provides more accurate representation but requires more bits to store or transmit the data.

5.3 System Model and Threat Model

In this section, we first describe the system model of the proposed scheme and then explain the threat model considered.

5.3.1 System Model

Our system model includes three main entities as explained in Chapters 3 and 4: clients, aggregators, and a server. In this work, clients are the entities of our proposed system whose data privacy we aim to protect.

5.3.2 Threat Model

In our threat model, we consider a set of threats that includes several possibilities for energy theft attacks and feature inference attacks.

Energy Theft Attacks (ETA)

In these attacks, we consider that a percentage of clients may modify their consumption/generation data with the goal of gaining a monetary advantage. This means that an energy theft attack can be launched by either increasing the generation readings to earn extra money or decreasing the consumption readings to lower the bill. Therefore, any deviation from the actual value of both generation and consumption readings is considered energy theft. This type of attack not only causes monetary losses but also greatly affects the safety and accuracy of the whole system's operations. A great number of demand management approaches rely on the data sent by the clients. If there is an energy theft of any kind, it will affect the accuracy and reliability of any estimated future demand.

We consider the four types of energy theft attacks introduced previously in Chapter 3, which are: consumer thefts, prosumer thefts, consumers balance thefts and prosumers balance thefts. We also consider an additional case where a balance attack is achieved by a single client. In this case, a prosumer may wish to reduce their reported consumption by a constant value or percentage for a period of time T and increase their own reported production by the same value. This is a new variant of the balance attack concept launched by a single user and we refer to it as *single-client balance thefts*.

Feature Inference Attacks (FIA)

As explained in Chapter 4, in a feature inference attack, the adversary tries to find attributes that are close to their actual values with a success rate significantly greater than a random guess. In this chapter, we extend the definition of FIA to include three different scenarios where passive adversaries try to build an inference model that maps the split layer outputs to the original raw readings of the client. In particular, the adversary follows these steps to perform the feature inference attack: 1) The adversary builds a dataset that contains their raw readings as targets and the split layer's output as features of this dataset. 2) The adversary builds an inference model $W_c^{(-1)}$ that has the opposite structure of the client's proposed model W_c and trains it to map the

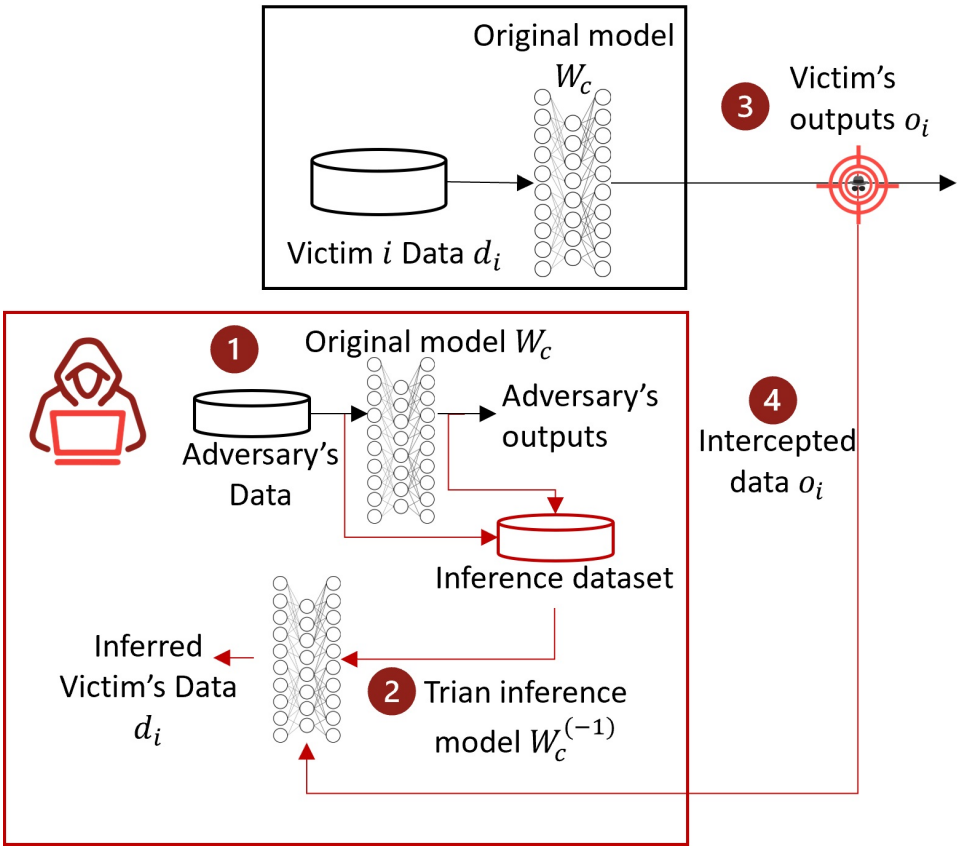


Figure 5.1: *Illustration of the Feature Inference Attack*

client's outputs o_i to their original raw data d_i. 3) A victim smart meter i reports their model's outputs o_i (i.e., the output of the split layer via running the forward pass of the victim's model) to the aggregator. 4) The adversary captures those data and tries to infer the original features from the split layer outputs using W_c^{-1}, the previously built inference model from step 2. The steps of this attack are illustrated in Figure 5.1.

As stated before, we study this attack under the following three sets of assumptions:

- **Case 1 (FIA1):** In this adversary setting, we assume that the adversary is a client with a dataset that comes from the same distribution as the victim's training data. Also, since both the victim and the adversary are clients, they have the same model structure. The adversary can use this to their advantage to build the inference model. This is the most strict but realistic attack setting out of the three FIAs.

- **Case 2 (FIA2):** For this setting, we assume that two or more clients collude with each other in an attempt to train a more robust inference model. The inference model is then used to infer features from the split layer outputs of other victims. This adversary setting is more powerful than the previous one as it involves training the inference model with more data, allowing faster convergence.

- **Case 3 (FIA3):** This is another collision attack that includes the aggregator along with the clients. In this setting, the colluding clients train the inference model using their joint datasets, and the aggregator helps to guess the mask as it already knows the sum of all clients' masks.

Note that the proposed feature inference attack can be performed during the training or deployment phases of the proposed model.

5.4 Proposed Privacy-Preserving Scheme

Our proposed privacy-preserving scheme is a multi-output neural network model that takes every client's reading at time t and outputs three results: an indication of whether theft is suspected or not; an estimation of the energy theft value, i.e. the deviation from the actual reading (either an increase or a decrease in the production or consumption reading); and the estimated demand of the next period $t+1$. In this part, we show how we use split learning along with masking as privacy protection measures to protect the privacy of the client's energy data in the multi-output model. To protect clients' privacy from semi-honest aggregators and eavesdroppers, we employ split learning as the first layer for privacy protection. Moreover, we add an extra privacy-preserving measure, i.e. masking, to split learning to protect the privacy of the client's level outputs and prevent feature-inference attacks. We propose to use a masking-based privacy-preserving scheme that uses pseudorandom number generators to generate masking matrices that mask the clients' outputs of the split layer. We assume that the masking process is done in a trusted execution environment at the client device. This can ensure that the client cannot manipulate the stored random parameters and compromise the integrity and security of the system. The proposed scheme consists of *three* phases: *initialization*, *mask generation and verification*, and *privacy-preserving energy theft detection and demand estimation phase*. A summary of the notations is provided in Table 5.1. For simplicity, we omit the subscript k for every timestamp t as all operations are done in a single timestamp.

Table 5.1: *Notations*

Notation	Description
SM_i	Smart meter i
Ag_i	Aggregator i
M	System's modulus
A_i	The multiplier for the pseudorandom number generator algorithm of smart meter i
C_i	The increment for the pseudorandom number generator algorithm of smart meter i
m	# of split layer outputs of a client
n	# of smart meters in a cluster
r_i	The vector of random numbers used by smart meter i
r_{ji}	A random number to mask the jth output of smart meter i
d_i	The readings (data) of smart meter i
o_i	The outputs of the split layer of smart meter i
o_{1-n}	The sum of clients 1 to n outputs
o_a	The output of the aggregator
$\hat{o}_i$	Masked outputs of smart meter i
$h(.)$	A hash function
$h^*(.)$	A homomorphic hash function

5.4.1 Initialization Phase

This phase consists of the following steps:

Step I1: During the initialization of the system operations, the server chooses a modulus $M \gg 0$ to be used during all system operations. Moreover, each client's smart meter SM_i is initialised with random parameters A_i and C_i to be used in the pseudorandom number generator algorithm, where the multiplier A_i should be $0 < A_i < M$ and the increment C_i should satisfy $0 \leq C_i < M$. These random parameters are stored on the server side and should be unique for every client. This is because we want to reduce the possibility of generating the same random numbers and recovering the original information in cases where passive adversaries intercept the communication.

Step I2: The server initialises each smart meter SM_i with a vector of random numbers of size $1 \times (m \times t)$ to be used to mask the m outputs of the split layer for t timestamps. This random vector is also stored at the server side for every SM_i, which makes the server store a $(n \times (m \times t))$ matrix at its side. Each random number in the matrix is denoted as r_{ji} for each $j \in 1...m$ and $i \in 1...n$. The value of the

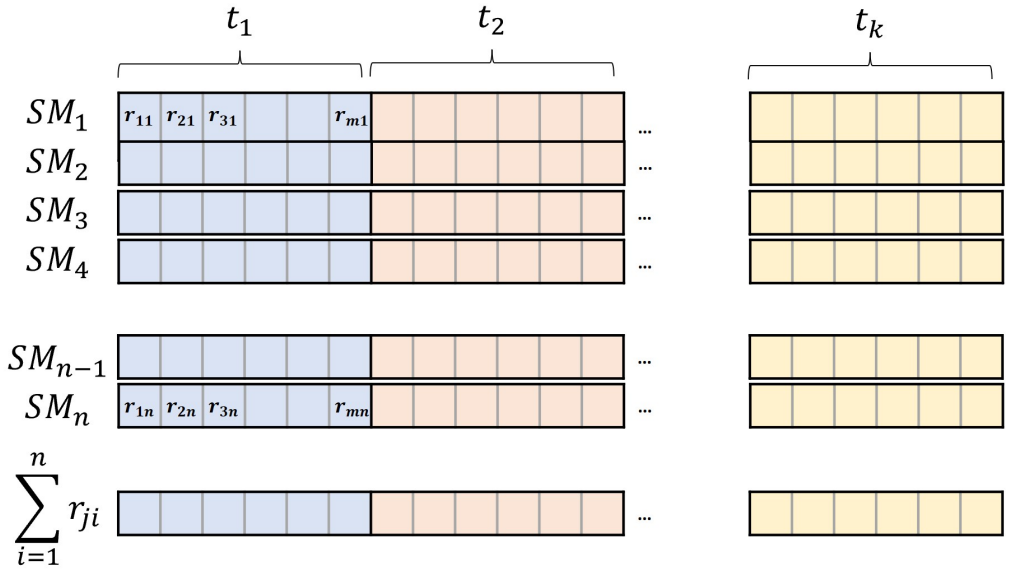


Figure 5.2: *Masking Matrix*

random numbers should be much larger than the outputs of each neuron to ensure that sensitive information is not leaked. The full matrix stored on the server side can be seen in Figure 5.2, where each row belongs to a single smart meter. We assume that these random numbers cannot be manipulated by the clients. This is because the manipulation of these parameters could potentially lead to the compromise of the whole masking approach.

Step I3: The server sends the summation of all random masks $\sum_{i=1}^{n} r_{ji}$, $\forall j \in 1...m$ to the aggregator using a secure channel to be used later in the unmasking process.

Step I4: The multi-output model is split between the system's entities, where every smart meter and every aggregator has a copy of the initialised version of its part. The proposed model is built as a stacked autoencoder (SAE), which is built by stacking multiple autoencoders to extract features layer by layer to obtain deeper and more abstract features that transform sensitive information into non-sensitive abstract data [138, 139]. SAEs have also been proven to be better at producing features than the traditional deep auto-encoders [140]. The distinct splits of the clients, aggregators and server can be seen in Figure 5.3.

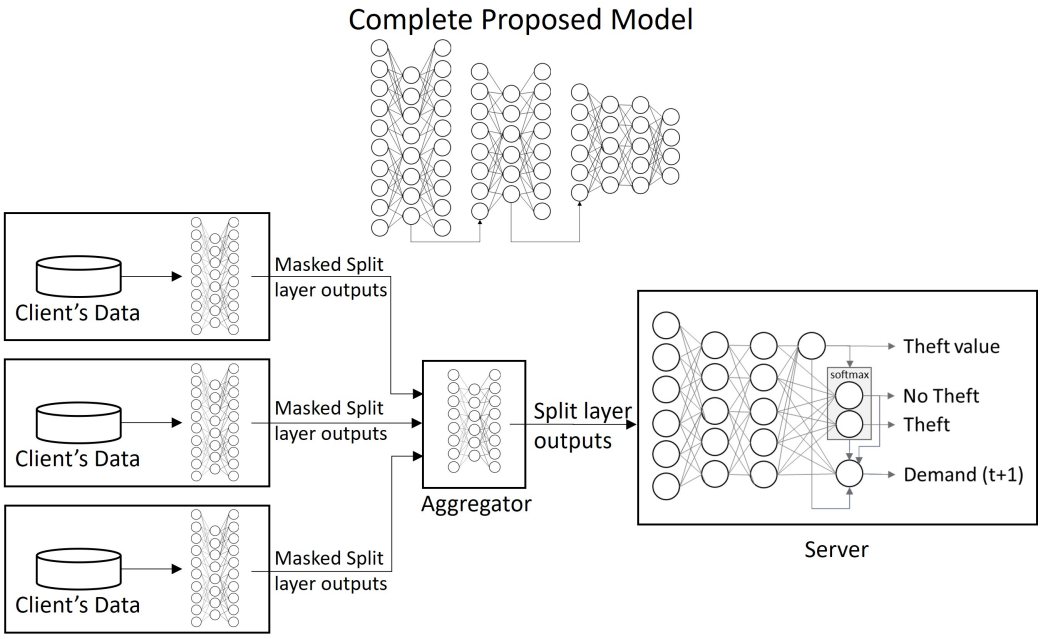


Figure 5.3: *The Multi-Output NN Model Architecture*

5.4.2 Mask Generation and Verification Phase

After t timestamps, the mask vectors and matrix get updated when all the random numbers are used. This is done during the mask generation and verification phase following these steps:

Step M1: Each SM uses its pseudorandom generator parameters A_i and C_i along with the linear congruential generator (LCG) algorithm to generate a new vector of random masks as follows: $r^*_{ji} = (A_i(r_{ji}) + C_i)\ mod\ M)\ ,\ \forall j \in 1...m$ and $\forall i \in 1...n$, where r^*_{ji} refers to the new random mask.

Step M2: At the server side, each client's vector is updated using the LCG algorithm and each client's unique pseudorandom parameters.

Step M3: The client calculates a hash integrity output $v_i = h(r^*_{ji} \,|\, t \,|\, SM_{id})$ using a one-way hash function and sends v_i to the server to acknowledge that they have the same set of random numbers.

Step M4: The server validates its sets of random numbers and sends an acknowledgement to the clients to confirm the new set.

Step M5: The summation of each new random mask $\sum_{i=1}^{n} r^*_{ji}\ ,\ \forall j \in 1...m$ is sent by the server to the aggregator using a secure channel.

5.4.3 Privacy-Preserving Energy Theft Detection and Demand Estimation Phase

This is the main phase of the proposed model, where the proposed multi-output model is trained in a privacy-preserving approach. This phase consists of the following steps:

Step P1: After the client and the server approve the masking matrix, each client uses their smart meter's energy reading for timestamp t to train their part of the model up to the split layer where they get the outputs o_i.

Step P2: To protect the privacy of these outputs, the client uses the random vector r_i to mask them. However, since the masking is carried out in the integer domain and the client outputs are of floating point numbers, the client needs to quantise the outputs first before masking them. Quantisation, as discussed in Section 5.2.2, is mainly done to map floats to integers. Here, it is done by mapping the min/max of the outputs (weights or activations) with a chosen min/max threshold of the integer range $[-\beta, \beta]$. Therefore, the outputs of the split layers o_i are first mapped to an integer in the $[-\beta, \beta]$ domain by performing the following: $o_i = truncate(o_i \times \beta)$ and then masked with the random vector using $\hat{o}_i = (o_i + r_i)\ mod\ M$. To ensure the validity of the masked data during transmission, the client calculates a verifier message v_i using a homomorphic hash function h^* and sends it, along with the masked data, to the aggregator. This is done as $v_i = h^*(o_i + r_i)$.

Step P3: After receiving the masked data and verifier messages from its set of clients, the aggregator aggregates the masked data $\sum_{i=1}^{n} \hat{o}_i$. Subsequently, the aggregator needs to unmask the aggregated data by subtracting the summation of the masks. This is done to obtain the unmasked outputs by $o_{1-n} = (\sum_{i=1}^{n} \hat{o}_i - \sum_{i=1}^{n} r_i)\ mod\ M$. Next, the aggregator needs to verify that the obtained unmasked output is correct and that the clients did not manipulate the masking process. Since the aggregator knows the summation of all random masks, it can calculate the verifier as $v^* = hash^*(o_{1-n} + \sum_{i=1}^{n} r_i)\ mod\ M$ and compare it with the aggregated verifiers $\sum_{i=1}^{n} v_i$. If the two values are equal, the obtained output is deemed correct. Next, the average of the obtained outputs $Avg(o_{1-n}) = o_{1-n} \div n$ needs to be dequantised using the opposite quantisation operation to use the average of the clients' outputs in the rest of the model.

Step P4: After that, the aggregator completes its part of the model to get its output o_a and sends these outputs to the server.

Step P5: Upon receiving all aggregators outputs at the server side, the server aggregates all received outputs and completes training the ML model. The final output of the server consists of mainly three outputs: (a) whether each client's input

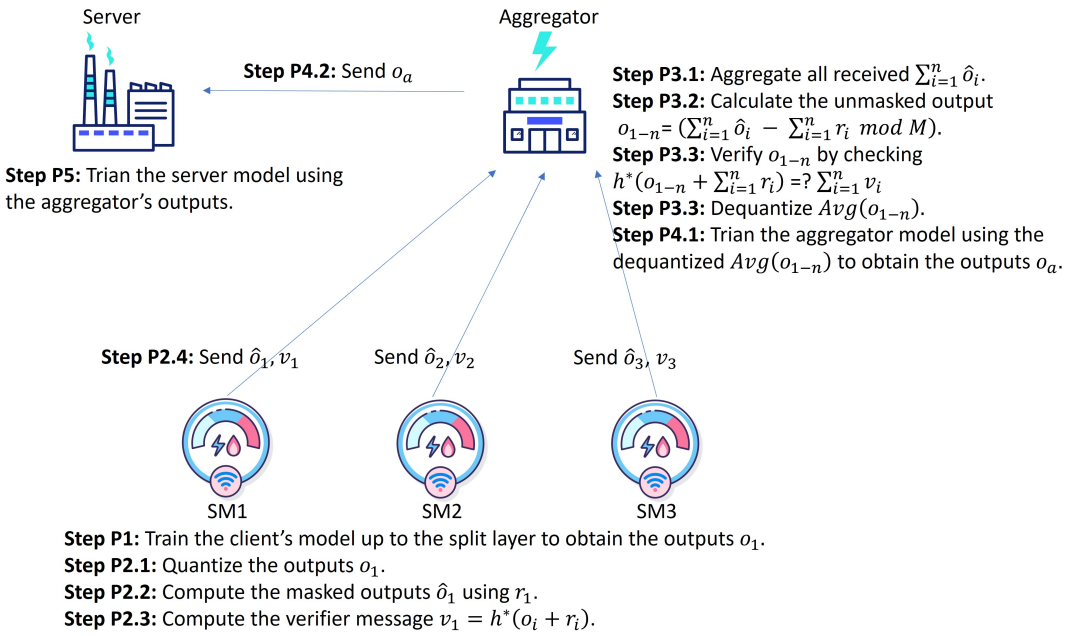


Figure 5.4: *Representation of the Privacy-Preserving Energy Theft Detection and Demand Estimation Phase Steps*

is an energy theft or not. (b) An estimation of the energy theft value. This output estimates how much the reported consumed or produced energy deviates from the actual ones. (c) And the final output estimates the energy demand for the next timestamp $(t + 1)$. The steps of this phase are illustrated in Figure 5.4

Privacy-Enhanced Energy Theft Detection and Demand Estimation Scheme

We expand our privacy-preserving multi-outputs model (shown in Section 5.4.3) and propose a more enhanced privacy scheme that trains an additional noisy layer at the client side. The trained client's part of the SAE extracts abstract features from the user's raw data. A small perturbation is added to these abstract features to maximise the independence between these abstract features and the raw data. This is done by training an extra layer at the end of the client's part of the model that takes the client's output as input and outputs the client's output with added Gaussian noise. The loss function of this noisy layer is the distance correlation $dCor$ between the raw data and the noisy output, and the training objective is to minimise it as much as possible. The steps of this scheme are as follows:

Step PE1: Define a noisy neural network layer with an input and output size of m, the same size as the output of the client's split layer.

Step PE2: Process the client's raw data through the client's model up to the split layer to obtain o_i.

Step PE3: Generate a set of Gaussian noise values and input it to the noisy layer to get a set of noise values added to o_i before they are masked.

Step PE4: Train the noisy layer to add Gaussian noise to the client's output in a way that minimises the $dCor$ between the raw data and the noisy outputs. In the training phase, two loss functions are calculated and a weighted combination is determined: the first loss function $L1$ is the original loss of the SAE at the server side that is responsible for accurately detecting energy thefts and estimating the theft's amount and energy demand, while the second loss function $L2$ minimises the $dCor$ value between the raw data and the noisy outputs. Since the two losses have conflicting objectives (accuracy vs. privacy), we need to combine the losses into one total loss. The total loss function can be calculated as follows: $LossFunction = L_1 + \alpha L_2$, where α weights the trade-off between accuracy and privacy. The rest of the scheme is performed in a similar way to those in Section 5.4.3 Step P2.

5.5 Experimental Setup

To evaluate our proposed scheme, we built our multi-output neural network using multiple Python 3 [106] libraries: Pandas [141], PyTorch [124] and Optuna [142]. We used Pandas for preprocessing the data, PyTorch for implementing the three-tier split learning architecture, and Optuna was used as a hyperparameter optimization framework to automate the search for the optimal hyperparameters for our proposed scheme. The search space included four different hyperparameters, including the number of hidden layers (between 6 and 20 inclusive), each hidden layer size (between 4 nodes and 128 nodes inclusive), the learning rate (between $1e^{-1}$ to $1e^{-5}$ inclusive), and the optimizer algorithm (either Adam, SGD or RMSprop). This search space was evaluated with a multi-objective function trading off maximising the accuracy of energy theft detection with minimising the mean squared error of both the estimated energy theft and the estimated demand. Preliminary trials using Optuna furnished us with the following effective hyperparameters used throughout our experiments: 10 hidden layers of sizes $[65, 91, 33, 89, 72, 33, 76, 30, 56, 44]$, a learning rate of $1e^{-4}$ and with the Adam optimizer. The ten hidden layers are split between the three tiers: client, aggregator, and server, as three layers, three layers, and four layers, respectively.

For our experiments, we reused our own developed dataset from Chapter 3 which includes energy readings of 1596 clients, 49 of which are prosumers with solar panels. For each client, 14 physical features along with 16 dynamic features are reported every 15 minutes. Since the readings in this dataset are all true (normal) readings, we mathematically changed 50% of them to be malicious data points. The attack scenarios described in Section 5.3.2 are used to produce these malicious points where constant deviations l are randomly chosen between 100 and 400 watts, and percentage deviations k are also randomly chosen between 10% and 40% of the actual reading. The dataset was split into 80% training and 20% testing with a batch size of 128. All input features were normalised using the Min-Max scaler using the default range [0,1]. We also extracted each reading's minute, hour, month, day, day of the year, and day of the week as extra features from the timestamp.

5.6 Results and Discussion

We conducted several experiments to cover the two broad threats identified in our threat model. In the first set of experiments, we evaluate the accuracy of our proposed model in detecting energy thefts and estimating theft values and energy demand. This set of experiments highlights results in cases of the five different energy theft attacks explained in our threat model. In the second part of this section, we evaluate the privacy of our proposed model and the privacy-enhanced version in terms of distance correlation and the successfulness of the three feature inference attacks (*FIA1* to *FIA3*) explained in our threat model. Results of the above two sets of experiments are presented in Section 5.6.1 and Section 5.6.2, respectively. Furthermore, we analysed the proposed scheme's computational overhead to ensure its applicability in real-world scenarios. Finally, in Section 5.6.4, we compare the proposed privacy-preserving approach with state-of-the-art approaches to give a fair view of where our proposed scheme stands.

5.6.1 Energy Theft Detection Experiments

The results of our energy theft attack detection and energy theft value and demand estimations are promising. We evaluated the performance of our system using several metrics, including accuracy, precision, recall (detection rate DR), and F1 score. We also used the coefficient of determination r^2 and the symmetric mean absolute error $SMAPE$ with a 95% confidence level to evaluate how good our estimations are for both estimating the theft value and the demand for the next timestamp ($t+1$). These

results are shown in Table 5.2, where we reported the results of every energy theft attack type presented in Section 5.3.2 on a single column. Then all attack types are presented together in a single dataset in the last column, *"All Theft Types"* column. The table also shows the r^2 and $SMAPE$ results for our two output estimations.

From the results in Table 5.2, we can see that our detection model performs exceptionally well in detecting all types of energy theft attacks. The overall performance is very good in the case of "All Theft Types" with an accuracy and F1 score of about 94% for both, a precision of 96.19%, and a detection rate of 92.17%. Moreover, our model performs well in estimating the energy theft values and the demand for $(t + 1)$, which is also illustrated in Figure 5.5. The table shows that the r^2 of the estimated theft value and the demand $t + 1$ ranged between 0.99 and 0.82, indicating good performance. The SMAPE values from the table show the percentages of the mean absolute error for our two estimates. We can see that the average percentage of error in estimating the theft value of "All Theft Types" is equal to 8.83%$\pm$0.56 with a 95% confidence level. In particular, in the event when all theft types are present in the dataset, our theft value estimates are between (the actual value - 8.83%) and (the actual value + 8.83%), i.e. Actual value $-$ 8.83% $\leq$ Predicted value $\leq$ Actual value + 8.83%. The table also shows that the SMAPE of estimating the theft value of prosumers' balance thefts is higher than the other results. This can be due to the specific characteristics of electricity production. Produced energy data, unlike consumption data, are sparse and irregular. For example, where solar panels are concerned, energy is produced only during daylight, and is zero otherwise. Moreover, energy production is extremely affected by weather conditions. Irregularity in the data and having gaps can lead to inaccurate estimations and thus skew the SMAPE value.

Furthermore, the estimates of the future demand for $(t + 1)$ are better with an absolute percentage error of 8.10% for the "All Theft Types" case. These results are also confirmed in Figure 5.5a and Figure 5.5b, where the actual values of theft (fraudulent deviation of consumed/produced energy) and demand are plotted against the predicted ones. From the two sub-figures, we can see that the model is performing well. The predicted values are significantly close to the regression line (i.e., the actual values), indicating low percentages of error.

To support our main motivation, which states that managing the demand-response of energy needs to take energy theft detection and estimation into account, we performed an experiment where we compared the performance of our demand estimation output in two cases: (a) taking theft detection and theft value estimation outputs into account and (b) in the case where the previous two outputs do not contribute to the demand $(t + 1)$ estimation output. The results of the two cases can be seen in Figure 5.6, where it is observable that considering theft detection and

Table 5.2: *Numerical Results of the Proposed Scheme for Different Energy Theft Attacks*

	Consumer Thefts	Prosumer Thefts	Consumers Balance Thefts	Prosumers Balance Thefts	Single-Client Balance Thefts	All Theft Types
Accuracy	99.99	99.86	99.83	99.01	99.67	94.46
Precision	99.99	99.85	99.79	98.97	99.34	96.19
Recall (DR)	99.98	99.86	99.85	98.95	99.99	92.17
F1 Score	99.99	99.85	99.82	98.96	99.66	94.14
r^2 (Theft Value)	0.99	0.99	0.99	0.99	0.95	0.92
r^2 (Demand $t+1$)	0.86	0.84	0.82	0.85	0.85	0.84
SMAPE (Theft Value)	2.48±0.04	2.89±0.14	3.89±0.07	26.63±0.64	8.39±0.13	8.83±0.56
SMAPE (Demand $t+1$)	7.57±0.07	7.60±0.05	8.30±0.07	7.87±0.05	7.91±0.07	8.10±0.06

theft value estimation when estimating future demand has a great advantage. There is an improvement of around 7.7% in how well our estimator estimates the actual demand values (i.e. r^2 value) and a significant decrease in the symmetric mean absolute percentage error ($SMAPE$) of the demand's estimates from 17.77 ±0.11 to 8.10 ±0.06. This confirms that taking theft detection and the estimated theft values into consideration in demand-response management is beneficial and justified.

Remark. *It is important to consider thefts and their values in managing the future demand for an energy system.*

5.6.2 Privacy Experiments

To assess how much our proposed model enhances the privacy aspect of the detection approach, we used two different sets of evaluation metrics. The first is by using distance correlation, defined in Section 4.2.1, and the other is by measuring the inference error of an inference attack. The inference error shows the degree of accuracy in inferring the private raw features, where higher errors indicate a lower likelihood of successfully launching a feature inference attack. These two metrics are assessed in the following two subsections.

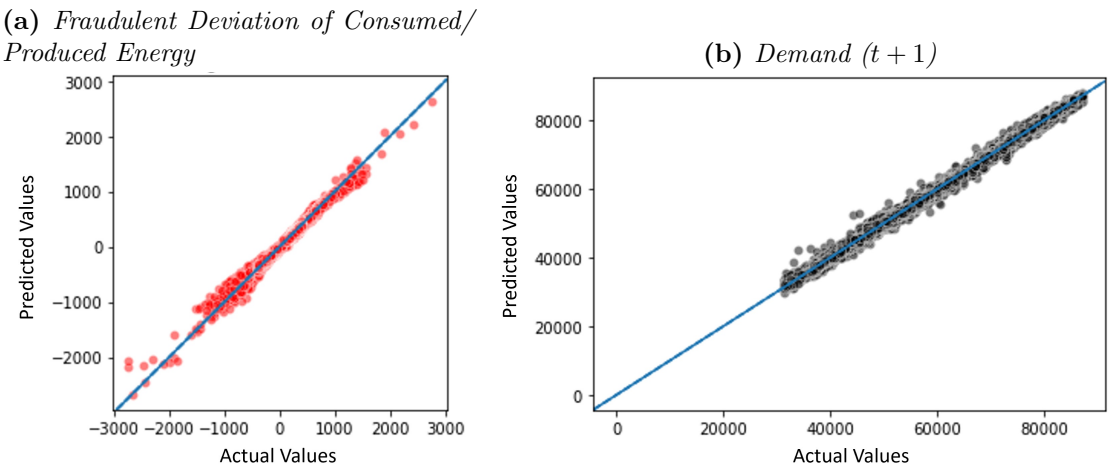
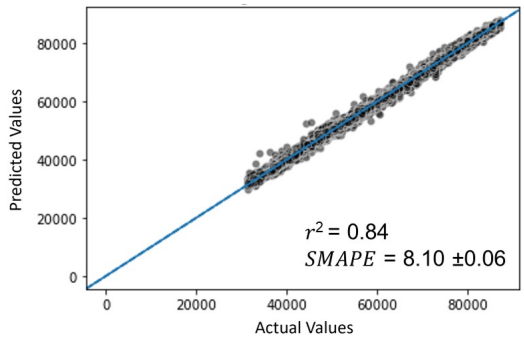


Figure 5.5: *Actual vs. Predicted Values of the Model's Two Outputs: Demand $(t+1)$ and Theft Value (Fraudulent Deviation of Consumed/Produced Energy)*

Privacy Analysis Using Distance Correlation

We evaluate the distance correlation $dCor$ between the users' inputs and the outputs that they send to the aggregator in an attempt to measure both linear and non-linear dependencies between the two. The aim is to lower this dependence as much as possible so that it would be difficult for an attacker to launch a successful feature inference attack. The first part of Table 5.3 compares the results between the non-privacy-preserving approach and the proposed privacy-preserving one using different masking levels. Each row indicates a different case where we used different values for β, which is the quantisation limit. As can be seen from the table, the $dCor$ value in the non-privacy approach is equal to 0.801, which is high, indicating a strong dependency between the private SM's inputs and the sent outputs. Then when we apply the privacy-preserving approach, the $dCor$ value decreases to 0.612, improving the privacy levels of the client's private reading by about 23%. Also, when setting β to the maximum level, we get almost the same detection performance results as the non-masking case, with a huge reduction (around 23% decrease) of the $dCor$ value between the user's input data and the outputs sent to the aggregator. The table also shows that the detection performance is unreliable when we set β to values less than $10e3$. The reason is that quantising a float to an integer with small ranges results in a huge precision loss which leads the model to be unable to learn how to detect theft accurately. Therefore, we adopt the value of $\beta = 10e8$ for all future experiments where we apply our proposed privacy-preserving approach.

(a) *Performance of estimating demand $(t+1)$ taking theft detection and theft value estimation into account*



(b) *Performance of estimating demand $(t+1)$ without taking theft detection and theft value estimation into account*

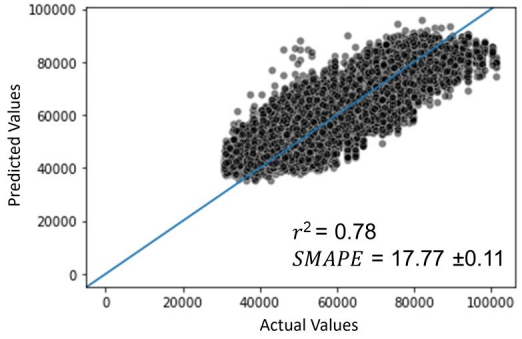


Figure 5.6: *Performance of Demand-Response Management Part of the System in Two Cases: (a) Taking Theft Detection and Theft Value Estimation into Consideration, and (b) Without Taking Them into Consideration*

The second part of Table 5.3, shows distance correlation results when the privacy-enhanced version of our proposed scheme is used. As described in Section 5.4.3, we add a noisy layer to the proposed approach to help conceal the users' inputs. The table shows how adding this noisy layer with different α values improves the distance correlation results. However, we also see that by setting α too large, the detection performance decreases as the machine learning model tries to optimise the distance correlation more. Setting α to a small value of 0.0001 will still give the same detection performance with increased privacy preservation of around 35% compared to the non-privacy approach.

Remark. *There is a clear trade-off between privacy and detection performance. The better privacy degree we achieve from lowering the dCor, the worse the results are in terms of detection accuracy.*

Privacy Analysis Using Inference Error

In this set of experiments, we evaluate the attacker(s)' abilities to launch a successful feature inference attack (FIA) against our proposed schemes. We compare how good the attacker is in inferring the victims' original data after building an inference model where these experiments are done as follows: the attacker(s) build an inference model W_c^{-1} by training an inverted version of the client's original model W_c. This inference model W_c^{-1} takes the split layer outputs o_i or the masked output $\hat{o}_i$ as inputs and

Table 5.3: *Performance of Different Masking Levels β and Different Noisy Layer Training Levels α*

Scheme Used		Accuracy	Precision	Recall (DR)	F1-Score	dCor
Non-privacy-preserving approach (no masking)		93.00	92.75	92.73	92.74	0.802
Privacy-preserving proposed scheme (with masking)	$\beta = 10e8$	94.46	96.19	92.17	94.14	0.612 (-23%)
	$\beta = 10e7$	93.44	93.56	92.79	93.17	0.613
	$\beta = 10e6$	93.05	93.19	92.35	92.77	0.613
	$\beta = 10e5$	92.28	90.48	93.88	92.15	0.613
	$\beta = 10e4$	91.86	91.11	92.10	91.60	0.667
	$\beta = 10e3$	89.36	86.81	91.92	89.29	0.692
	$\beta = 10e2$	57.53	53.36	94.91	68.31	0.797
	$\beta = 10e1$	55.39	52.08	94.25	67.08	0.801
Privacy-enhanced proposed scheme (with masking & noisy layer)	$\beta = 10e8$ $\alpha = 0.01$	77.45	75.33	80.24	77.42	0.238 (-70%)
	$\beta = 10e8$ $\alpha = 0.005$	86.23	90.13	82.45	85.21	0.241 (-70%)
	$\beta = 10e8$ $\alpha = 0.001$	88.64	93.75	82.11	88.30	0.295 (-63%)
	$\beta = 10e8$ $\alpha = 0.0005$	89.66	95.13	82.43	88.29	0.289 (-64%)
	$\beta = 10e8$ $\alpha = 0.0001$	91.14	95.41	90.33	89.23	0.525 (-35%)

trains the model to map them to their original data d_i. The inference model is trained using the attacker(s)' own data, which means that the more clients that collude to train the inference model, the more powerful it is. The model is tested to infer other victims' data from the outputs they send where these victims' data are not part of the training phase. After that, the mean squared error (MSE) between what has been inferred and the actual data is measured to assess the feature inference attack accuracy rate, where lower values of MSE indicate higher chances of attack success.

FIA1 Experiment In our first experiment, we look at the first type of feature inference attacks, *FIA1*, where one attacker builds an inference model and tries to infer other victims' original data. In Figure 5.7, we measured the average MSE of the inference model built in three cases: when the non-privacy-preserving approach (no

masking) is used, when the proposed privacy-preserving approach (with masking) is used, and finally when the proposed privacy-enhanced approach (with masking and noisy layer) is used. As we can see from the figure, the inference error is much less when the non-privacy-preserving approach is applied. The error is an average of $\sqrt{0.022} = 0.14$, which is significantly lower than the error in cases where one of our proposed privacy-preserving approaches is applied. With the proposed privacy-preserving and the privacy-enhanced approaches, the error is big in the first training rounds with an average value of $\sqrt{0.2} = 0.44$. This means that the inferred values have a mean error of 0.44, which is very high considering that all of our original raw feature values are normalised in a range between [0-1]. At later stages of the training, after 50 epochs, the error drops to around $\sqrt{0.051} = 0.22$, which is still double the error of the non-privacy-preserving approach.

Remark. *Using one of the proposed privacy-preserving approaches doubles the error of the feature inference attack making the attack less successful.*

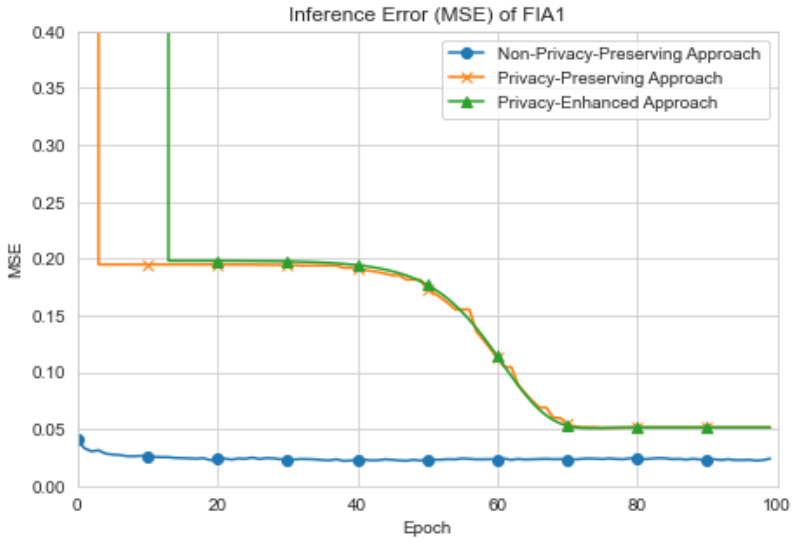


Figure 5.7: *Inference Error of Inference Attack FIA1 Using the Non-Privacy-Preserving Approach, the Proposed Privacy-Preserving Approach and the Proposed Privacy-Enhanced Approach*

FIA2 Experiment In the second experiment, we assess the inference error in case two or more clients collude with each other, which we refer to as *(FIA2)*. In this attack, the colluding attackers will train an inference model using their joint datasets. The built model is used to infer other victims' original data (those clients who did not participate in the training). Figure 5.8 shows the results of the inference error in

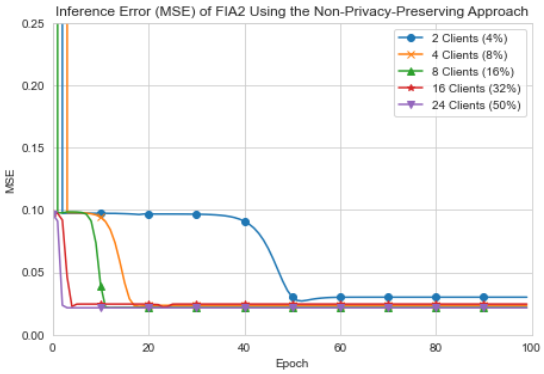
terms of MSE in three cases: (a) using the non-privacy-preserving approach; (b) using the proposed privacy-preserving approach; (c) using the proposed privacy-enhanced approach; and finally (d) a summary figure comparing the three approaches together. In all cases, there were 49 clients in the same cluster, and we tested with different percentages of colluding clients. The first thing we notice from these figures is that as the number of colluding attackers increases, the inference error drops faster. This confirms the basics of any machine learning where having more data in training results in faster convergence and more accurate models [143].

Looking at Figure 5.8a, we can see the results of performing an *FIA2* attack in the case where the non-privacy-preserving approach is used. Once again, we can see that the inference model converges faster and gives better results as the number of colluding clients increases. Moreover, in all the cases of different colluding client numbers, the MSE of the inference attack is at around 0.022, which is almost half the error in cases where either the proposed privacy-preserving approach (shown in Figure 5.8b) is used or when the privacy-enhanced version (shown in Figure 5.8c) is used. To summarise these results, the last sub-figure, Figure 5.8d, shows a comparison between the inference error of attack *FIA2* using the non-privacy-preserving approach and the two proposed privacy-preserving approaches. In this figure, the MSE is used for this comparison after training the model for 100 epochs. We can see that the inference error of the proposed privacy-preserving approaches is more than the non-privacy approach for all different numbers of colluding clients. It is double the error in all these cases, indicating an added advantage of using the proposed schemes over the non-privacy one and making it difficult for attackers to launch accurate feature inference attacks even when they collide.

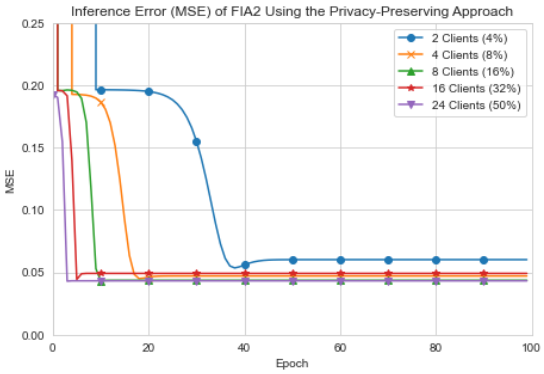
Remark. *An increased number of colluding attackers in an FIA allows those attackers to get a more accurate inference model faster.*

FIA3 Experiment In this experiment, we evaluate the attackers' ability to perform the last feature inference attack, *FIA3*. In this attack, a group of malicious clients collude with an aggregator. We performed this attack in case our proposed privacy-preserving approach is used and compared it with the results of FIA2. From Figure 5.9, we see that there is not much advantage of having the aggregator as a collaborator in this attack. This proves that splitting our proposed privacy-preserving scheme between the system's entities improves the system's privacy even with honest-but-curious aggregators.

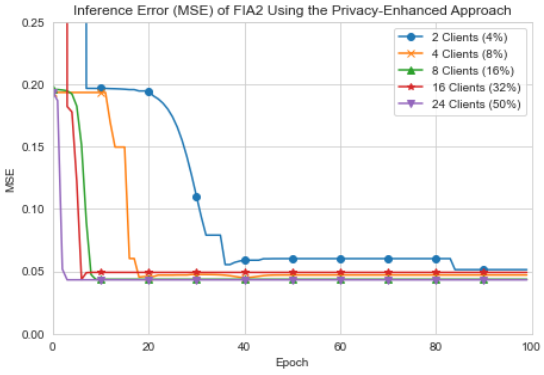
(a) *Non-privacy-preserving approach*



(b) *Privacy-preserving approach*



(c) *Privacy-enhanced approach*



(d) *Summary comparison between the three approaches*

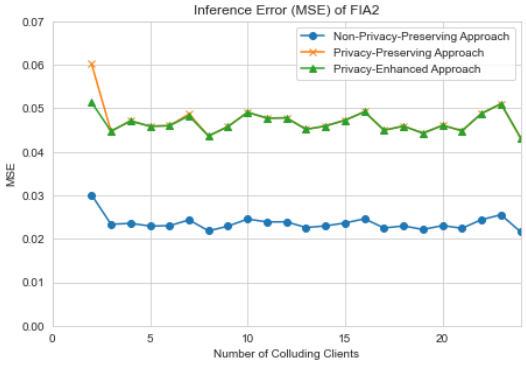


Figure 5.8: *Inference Error of FIA2 Using the Non-Privacy-Preserving Approach, the Proposed Privacy-Preserving Approach, and the Privacy-Enhanced One*

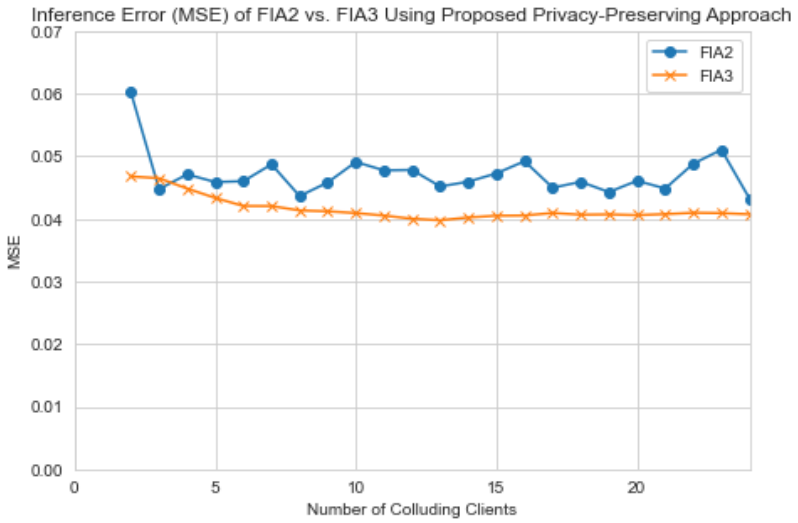


Figure 5.9: *Inference Error of FIA3 vs. FIA2 Using the Proposed Privacy-Preserving Approach*

5.6.3 Computational Overhead

In this section, we present the analysis conducted to evaluate the performance of our proposed models in terms of computational time. We do so by computing the time needed to complete the execution of one round at the client's smart meter. This analysis provides valuable insights into the efficiency of a model. Here, we compare three models: the privacy-preserving proposed scheme (with masking), the privacy-enhanced proposed scheme (with masking & noisy layer), and the federated learning scheme.

The results in Table 5.4 reveal significant differences among the three schemes. The privacy-preserving proposed model has the shortest execution time, as expected, due to its simplicity in providing privacy using masking and because only part of the model is executed on the client's side. The federated learning scheme exhibited the longest execution time of the three models. The fact that the complete model needs to be run in the smart meter resulted in more extensive calculations leading to longer execution times. Surprisingly, our privacy-enhanced proposed model has almost doubled the execution time of the privacy-preserving scheme. This indicates that the addition of only a single layer can lead to a notable increase in execution time.

Table 5.4: *Computational Overhead*

Scheme	Computational Time (in seconds)
Privacy-preserving proposed scheme (with masking)	0.0049
Privacy-enhanced proposed scheme (with masking & noisy layer)	0.0090
Federated Learning	0.0163

5.6.4 Comparison with Other Privacy-Preserving Schemes

To give a fair evaluation of our proposed model against state-of-the-art energy theft detectors, we compared our work with the previously reviewed privacy-preserving, federated-learning-based approaches: FedDetect [68] and FedDP [69]. Similar to these approaches, we used the State Grid Corporation of China (SGCC) dataset [53]. In particular, Table 5.5 shows the results of the three models in terms of accuracy, precision, recall, and F1-score. Note that the authors of FedDetect [68] report only the accuracy. As we can see from this table, the performance of the three approaches is relatively the same but with a great increase in the recall (detection rate) result from our proposed scheme. Therefore, we can argue that our approach gives better results in terms of energy theft detection. Moreover, although the performance of our proposed scheme is almost the same as [68] and [69] in terms of energy theft detection, these two schemes lack some important features, as discussed previously in Section 2.10 and Table 2.9. Both [68] and [69] do not provide functionalities for estimating the energy theft value, estimating the future demand, and preserving privacy with minimal communication and computation overheads.

5.7 Threats to Validity

In this section, we address threats to the validity of our multi-output energy theft detection scheme. These threats required careful consideration and discussion. One threat to validity concerns the architecture and hyperparameter settings of the multi-output NN model employed here. It is possible that other architecture choices, such as the depth of the NN model and the number of neurons in each layer, could produce different results than those reported here. This has been mitigated by our

Table 5.5: *Comparison of Previous Literature with the Proposed Scheme in Terms of Accuracy, Precision, Recall, and F1 Score*

Scheme	Accuracy	Precision	Recall	F1 Score
FedDetect [68]	91.90	-	-	-
FedDP [69]	91.67	89.03	91.67	88.72
Our Proposed Scheme	91.63	88.18	96.14	91.99

use of a hyperparameter optimisation tool that facilitated such choices. We used Optuna to conduct extensive model selection and hyperparameter tuning, allowing us to find very high-performing models for our simulation dataset. Another threat to the validity of our study arises from the simplifying assumptions made about the security of the masking process. This assumption is reasonable and can be easily implemented in real-world scenarios with the help of trusted execution environments.

There are threats to the generalisation of the proposed model's findings to real-world practice. To mitigate such threats and to provide unbiased and impartial results, we tested our proposed work using the real-world energy dataset (SGCC) and compared the results with two state-of-the-art models. Although this does not ensure the generality of our results, it is significant evidence that the outcomes are likely applicable to other datasets in the same field.

5.8 Summary

This chapter investigated the possibility of achieving our *third hypothesis* by proposing a privacy-preserving energy theft detection scheme joined with demand-response management for smart grid systems. The proposed scheme is the first to bridge the gap between the two issues. The accuracy of the proposed scheme's theft detection is analysed to confirm its robustness against five types of energy theft attacks. Moreover, the performance of the demand forecasting is also analysed in two settings and results suggest that considering thefts' magnitudes when forecasting future demand provides better results. In addition, two sets of privacy metrics are proposed and evaluated to ensure that the scheme provides individual meter readings' privacy. The overall conclusion of all the experiments shows that the results confirm the third hypothesis. The proposed scheme outperforms the existing privacy-preserving energy

theft detectors in terms of detection rate and has significantly greater capabilities than other approaches as it can estimate the amount of theft along with the future demand with high accuracy.

Chapter 6

Conclusion and Future Work

In this chapter, we summarise the main findings of this thesis work. The thesis hypotheses, provided in Chapter 1, are also revisited to highlight how the research conducted in this study supports them. Finally, open issues and future work are discussed.

6.1 Summary of Experiments

Energy theft attacks are considered one of the crucial attacks against smart grids. Small alterations in the sensed data measurements can severely disrupt these sensitive networks. Therefore, it is very important to employ defence schemes that counter them. Several attempts have been made to design such schemes, but significant limitations remain. The major problems with existing mechanisms for energy theft detection systems are:

- Different research groups tackle the problem of energy thefts from different perspectives. Electrical engineering academics deal with the problem using theoretical approaches such as state estimation techniques that rely heavily on sensor data from electrical grids. However, as we have seen from our literature review in Section 2.7, these techniques are complex, unscalable and do not consider the security of the communication networks. On the other hand, methods used by computer scientists, such as secure communication and encryption, can be resource-intensive.

- Existing solutions focus on a single type of attack scenario launched by *consumers*. However, in Section 3.1, we saw how the number of prosumers has risen significantly in recent years. The introduction of *prosumers* as new actors who participate in electrical grid operations must now be taken into consideration.
- The modernisation of the traditional electrical grids into sensor-based energy systems has expanded the scale of the produced data from different sources. However, existing work for detecting energy theft does not fully exploit these features.
- Privacy of customers' data is often overlooked. Even when considered, the privacy level of a solution is not evaluated. This has been highlighted in detail in Section 2.8.
- It has been shown from our review in Section 2.7 that the research in energy theft has focused solely on the detection part and has not extended beyond that. The effect of energy thefts on other functionalities of grids has not been studied.

To address these limitations, we develop three experimental studies and rigorously evaluate their performance. Our main objective for this work was to be able to detect a diverse range of energy thefts accurately and efficiently while preserving customers' privacy.

Machine learning and deep learning approaches are examined as a means of anomaly detection in this thesis. The first experimental study in Chapter 3 presented a *cluster-based* detector that is able to detect various *energy theft attacks from both consumers and prosumers*. We also developed the notion of a *balance attack*, a novel attack scenario in which attackers attempt to hide their thefts by balancing the entire net consumed and generated power. The simulations use a *generated dataset* that includes prosumers' and consumers' energy usage, as well as data from numerous data sources. The implementations and findings of this work were able to show a robust capability in detecting all of the eight considered energy theft scenarios launched by the two types of customers. The developed model was tested using different ML approaches in different scenarios, and all results showed high detection rates. This can confirm what we presented in *hypothesis 1 "Combining machine learning techniques (clustering and classification) can enhance the detection of a range of thefts, including prosumers thefts"*.

Chapter 4 considered the problem of *preserving privacy* in an ML-based energy theft detection context. We introduced *Three-Tier Split Learning* as a private

collaborative machine learning method for detecting energy thefts without the requirement to utilise raw data. Our experiments showed that our proposed model outperformed non-private ones, even in the presence of *poisoning attacks*. The approach exhibits strong privacy preservation, which was quantified by the *distance correlation* metric, without introducing much communication overhead. Hence, it provides both effective communication and privacy preservation. *Hypothesis 2 "A privacy-preserving ML technique that suits the smart grid environment can be developed to accurately and effectively detect energy theft while preserving the privacy of customers' data"* is thus supported with the results in Chapter 4.

The *integration between energy theft detection and demand management* topics is then studied in Chapter 5. The work developed is the first to provide a solution for both problems. We developed a privacy-preserving multi-output model that is able to detect energy thefts, estimate their magnitudes and use them to aid demand estimation management. Energy theft detection capabilities, the accuracy of future demand estimations and privacy preservation level were all tested. The results from our sets of experiments confirm and support our *third hypothesis "A multi-output neural network framework can be used to simultaneously predict the presence of theft, predict its magnitude, and use that estimation to make more accurate forecasts"*.

6.2 Future Directions

The research of effective detection and defence strategies against energy thefts seems to be an endless path. In this thesis, we have provided evidence for the specific hypotheses given in Section 1.1. Future investigation on the following points may fruitfully extend and build on the contributions made in this thesis:

Exploration of New Threats: In this thesis, we focused on a certain set of energy theft and privacy attacks. Further research is necessary to analyze and investigate other attacks. In future, it would be interesting to address ML-enabled threats, where attackers use machine learning to evade or manipulate the detection system. Attackers may use various techniques, such as adversarial attacks and generative adversarial networks (GANs), to deceive the detection system. Adversarial examples generated by ML-enabled threats can mimic the behaviour of normal energy consumption and thus evade detection. Moreover, it is crucial to develop countermeasures to mitigate these threats and ensure the security and reliability of energy systems.

Variation in Theft Strategy: Throughout this thesis, we adopted a straightforward but meaningful attack strategy for energy thefts where we had scenarios of different attack types and others with different magnitudes of thefts. However, the future direction can examine more sophisticated types of attack where attackers can vary their attack pattern *over time*. When to perform the attack and how often, the attack's duration, the theft's magnitude, and the type of theft can all be varied over time. An attacker may avail themselves of such flexibility to behave more stealthily and avoid detection. Widening the attack space in this way facilitates the examination of so-called advanced persistent threats (APTs) in smart grids. We believe that the introduction of such novel attack strategies in SGs requires considering the corresponding countermeasures. We also note that ML-enabled strategies (see above) may well be developed to make optimal use of available attack flexibility.

Privacy Attacks: In this thesis, particularly in Chapters 4 and 5, we considered two privacy attacks: poisoning attacks and feature inference attacks. However, as stated in Section 2.8, there are other ML-based privacy attacks that could be used to extract sensitive information from machine learning models, such as model extraction, model stealing, and label inference attacks. It is worth studying their applicability to smart grids.

Privacy Metrics: A privacy metric aims to measure the level of privacy gained in a system by applying a privacy-preserving technique. Despite the large number of metrics in the literature, quantifying privacy in energy theft research remains unstudied. In Chapters 4 and 5, we focused on a single metric. It would be interesting to examine the use of other measures of information leakage and formally structure a unified one that could be used for the evaluation of other privacy-preserving techniques.

Attack Response and Recovery: The primary focus of our research has been on the detection aspects of energy theft attacks by generating alerts. However, there is a trend in intrusion detection research to take active responses to effectively stop current threats, prevent future attacks and restore normal operations. In the energy theft detection research, this has not been discussed thoroughly and is a potential direction for future research. A developed response-aware system has to consider each situation's factors and constraints. This includes modelling the cost of response actions, considering factors such as time and resource constraints, as well as the degree of autonomy and human involvement.

Energy Theft Prevention: Our approach assumes that users are able to misreport, and we leave the mechanisms by which this is achieved unspecified. (This allows us to investigate detection approaches independent of the actual means of attack.) Clearly, making such misreporting more difficult, and preferably impractical, would be a major

advance. This will inevitably involve the development of tamper-resistant hardware in the grid systems (particularly at the user's end). The use of machine learning, in particular, to detect attacks on such hardware seems largely uninvestigated.

Adaptive Detection System: A limitation that most offline ML-based detection mechanisms face (including ours) is that customers' energy behaviours can fluctuate over time which makes the ML-based system susceptible to *model drift*. Model drift refers to the situation where the model predictions are degrading over time due to changes in the environments or consumption/production behaviour. Such change in the environment requires the model to be updated using online and adaptive learning processes. This is an important issue that needs full investigation in future research. The suggested approach is that instead of halting the learning process after the training phase, the algorithms should continue learning during the operation of the system. This enables the system to learn from the misclassified cases. By doing so, the system can reduce the occurrence of false positives and adapt to changing environments.

Development of a Real Comprehensive Dataset: Having a real-world and comprehensive energy dataset is an important area of research that can provide valuable insights into smart grid research in general and energy theft detection research in particular. Our approaches could be tested using such a dataset to ensure that the models are accurate, reliable, and effective in real-world scenarios. It also helps to ensure that the models are robust, generalize well, and can be used safely and ethically in practice.

Wider Exploration of the ML Pipelines: We have made specific choices with respect to the ML pipelines adopted in this thesis. There is likely benefit to be gained from adopting a wider set of techniques within the pipeline. For example, there are many data preprocessing and feature engineering approaches that could be adopted. A wider exploration of the ML pipeline involves considering the different approaches in these stages to identify opportunities that improve the overall process. In the data preprocessing stage, we can investigate different data cleaning and normalizing techniques. In the feature engineering stage, feature reduction techniques might explore and select features that are most relevant to the problem being solved. Another possibility in this direction is to develop a probabilistic classification model, which gives the attack's probability, rather than a straightforward (attack, not attack) one. This can aid the process of choosing the appropriate response for a detected attack based on that likelihood.

Finding the Perfect Balance Between False Negatives and False Positives: Most energy theft detection work seeks to optimise either the detection rate or accuracy of the detector. Neither guarantees that the number of false positives

or false negatives is sufficiently low to make a system practical. Having high false positive rates in an energy theft detector results in an increased number of unnecessary inspections that eventually will reduce the economic return of the theft detection. On the other hand, false negatives vary in their importance; larger theft magnitudes will generally be of more concern than small ones. Therefore, finding a practical balance between the two in the detection approaches should be investigated.

Bibliography

[1] Yang Liu, Ting Liu, Hong Sun, Kehuan Zhang, and Pengfei Liu. Hidden electricity theft by exploiting multiple-pricing scheme in smart grids. *IEEE Transactions on Information Forensics and Security*, 15:2453–2468, 2020.

[2] Zhongzong Yan and He Wen. Performance analysis of electricity theft detection for the smart grid: An overview. *IEEE Transactions on Instrumentation and Measurement*, 71:1–28, 2022.

[3] Patrick Glauner, Jorge Augusto Meira, Petko Valtchev12, Radu State, and Franck Bettinger. The challenge of non-technical loss detection using artificial intelligence: A survey. *International Journal of Computational Intelligence Systems*, 10(1):760, 2017.

[4] Kaile Zhou, Shanlin Yang, and Zhen Shao. Energy internet: the business perspective. *Applied Energy*, 178:212–222, 2016.

[5] Pardeep Kumar, Yun Lin, Guangdong Bai, Andrew Paverd, Jin Song Dong, and Andrew Martin. Smart grid metering networks: A survey on security, privacy and open research issues. *IEEE Communications Surveys & Tutorials*, 2019.

[6] Jeremy Rifkin. *The third industrial revolution: how lateral power is transforming energy, the economy, and the world*. Macmillan, 2011.

[7] Prosanta Gope and Biplab Sikdar. An efficient privacy-preserving authentication scheme for energy internet-based vehicle-to-grid communication. *IEEE Transactions on Smart Grid*, 10(6):6607–6618, Nov 2019.

[8] Yijia Cao, Qiang Li, Yi Tan, Yong Li, Yuanyang Chen, Xia Shao, and Yao Zou. A comprehensive review of energy internet: basic concept, operation and planning methods, and research prospects. *Journal of Modern Power Systems and Clean Energy*, 6(3):399–411, 2018.

[9] Divam Lehri and Arjun Choudhary. A survey of energy theft detection approaches in smart meters. In *Intelligent Energy Management Technologies*, pages 9–24. Springer, 2021.

[10] Xiaofang Xia, Yang Xiao, Wei Liang, and Jiangtao Cui. Detection methods in smart meters for electricity thefts: A survey. *Proceedings of the IEEE*, 110(2): 273–319, 2022.

[11] Ahlam Althobaiti, Anish Jindal, Angelos K Marnerides, and Utz Roedig. Energy theft in smart grids: A survey on data-driven attack strategies and detection methods. *IEEE Access*, 9:159291–159312, 2021.

[12] Ahmed S Musleh, Guo Chen, and Zhao Yang Dong. A survey on the detection algorithms for false data injection attacks in smart grids. *IEEE Transactions on Smart Grid*, 2019.

[13] Lei Cui, Youyang Qu, Longxiang Gao, Gang Xie, and Shui Yu. Detecting false data attacks using machine learning techniques in smart grid: A survey. *Journal of Network and Computer Applications*, page 102808, 2020.

[14] Yuning Jiang, Manfred Jeusfeld, Yacine Atif, Jianguo Ding, Christoffer Brax, and Eva Nero. A language and repository for cyber security of smart grids. In *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*, pages 164–170. IEEE, 2018.

[15] Abubakar Sadiq Sani, Dong Yuan, Jiong Jin, Longxiang Gao, Shui Yu, and Zhao Yang Dong. Cyber security framework for internet of things-based energy internet. *Future Generation Computer Systems*, 93:849–859, 2019.

[16] Haibo He and Jun Yan. Cyber-physical attacks and defences in the smart grid: a survey. *IET Cyber-Physical Systems: Theory & Applications*, 1(1):13–27, 2016.

[17] Ekram Hossain, Zhu Han, and Vincent Poor. *Smart grid communications and networking*. Cambridge University Press, 2012.

[18] Yasin Kabalci. A survey on smart metering and smart grid communication. *Renewable and Sustainable Energy Reviews*, 57:302–318, 2016.

[19] Accenture, Ponemon Institute LLC. The cost of cybercrime, ninth annual cost of cybercrime study. Technical report, Accenture, Ponemon Institute LLC, 2019. URL https://mysecuritymarketplace.com/reports/ninth-annual-cost-of-cybercrime-study/.

[20] Lars Fischer, Mathias Uslar, Doug Morrill, Michael Döring, and Edwin Haesen. Study on the evaluation of risks of cyber-incidents and on costs of preventing cyber-incidents in the energy sector. Technical report, Ecofys, 2018. URL https://www.offis.de/en/offis/publication/study-on-the-evaluation-of-risks-of-cyber-incidents-and-on-costs-of-preventing-cyber-incidents-in-the-energy-sector.html.

[21] Bill Miller and Dale Rowe. A survey SCADA of and critical infrastructure incidents. In *Proceedings of the 1st Annual Conference on Research in Information Technology*, pages 51–56, 2012.

[22] David Denkenberger, Anders Sandberg, Ross John Tieman, and Joshua M Pearce. Long-term cost-effectiveness of interventions for loss of electricity/industry compared to artificial general intelligence safety. *European Journal of Futures Research*, 9(1):1–24, 2021.

[23] Muhammed Zekeriya and Resul Das. Cyber-security on smart grid: Threats and potential solutions. *Computer Networks*, 169:107094, 2020.

[24] Zakaria El Mrabet, Naima Kaabouch, Hassan El Ghazi, and Hamid El Ghazi. Cyber-security in smart grid: Survey and challenges. *Computers & Electrical Engineering*, 67:469–482, 2018.

[25] Wei Yu, David Griffith, Linqiang Ge, Sulabh Bhattarai, and Nada Golmie. An integrated detection system against false data injection attacks in the smart grid. *Security and Communication Networks*, 8(2):91–109, 2015.

[26] Gautam Raj Mode, Prasad Calyam, and Khaza Anuarul Hoque. Impact of false data injection attacks on deep learning enabled predictive analytics. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7. IEEE, 2020.

[27] Sulabh Bhattarai, Linqiang Ge, and Wei Yu. A novel architecture against false data injection attacks in smart grid. In *2012 IEEE International Conference on Communications (ICC)*, pages 907–911. IEEE, 2012.

[28] Nemanja Zivkovi and Andrija T Sari. Detection of false data injection attacks using unscented kalman filter. *Journal of Modern Power Systems and Clean Energy*, 6(5):847–859, 2018.

[29] Meng Zhang, Chao Shen, Ning He, SiCong Han, Qi Li, Qian Wang, and XiaoHong Guan. False data injection attacks against smart gird state estimation: Construction, detection and defense. *Science China Technological Sciences*, pages 1–11, 2019.

[30] Qi Wang, Wei Tai, Yi Tang, and Ming Ni. Review of the false data injection attack against the cyber-physical power system. *IET Cyber-Physical Systems: Theory & Applications*, 4(2):101–107, 2019.

[31] Alvaro A Cárdenas, Saurabh Amin, Galina Schwartz, Roy Dong, and Shankar Sastry. A game theory model for electricity theft detection and privacy-aware control in AMI systems. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1830–1837. IEEE, 2012.

[32] Saurabh Amin, Galina A Schwartz, Alvaro A Cardenas, and S Shankar Sastry. Game-theoretic models of electricity theft detection in smart utility networks: Providing new capabilities with advanced metering infrastructure. *IEEE Control Systems*, 1(35):66–81, 2015.

[33] Longfei Wei, Aditya Sundararajan, Arif I Sarwat, Saroj Biswas, and Erfan Ibrahim. A distributed intelligent framework for electricity theft detection using benford's law and stackelberg game. In *2017 Resilience Week (RWS)*, pages 5–11. IEEE, 2017.

[34] Mohsin Ahmed, Abid Khan, Mansoor Ahmed, Mouzna Tahir, Gwanggil Jeon, Giancarlo Fortino, and Francesco Piccialli. Energy theft detection in smart grids: Taxonomy, comparative analysis, challenges, and future research directions. *IEEE/CAA Journal of Automatica Sinica*, 9(4):578–600, 2022.

[35] Muhammad Salman Saeed, Mohd Wazir Mustafa, Nawaf N Hamadneh, Nawa A Alshammari, Usman Ullah Sheikh, Touqeer Ahmed Jumani, Saifulnizam Bin Abd Khalid, and Ilyas Khan. Detection of non-technical losses in power utilities—a comprehensive systematic review. *Energies*, 13(18):4727, 2020.

[36] Rohit Grewal, Tushar Sharma, Rajbir Mourya, Anil Kumar, and Karamjit Kaur. Cost effective overload and theft detection for power distribution system. In *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 450–455. IEEE, 2018.

[37] Muhammad Saad, Muhammad Faraz Tariq, Amna Nawaz, and Muhammad Yasir Jamal. Theft detection based gsm prepaid electricity system. In *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)*, pages 435–438. IEEE, 2017.

[38] R Sathyapriya and V Jeyalakshmi. Hardware implementation of IOT based energy management theft detection and disconnection using smart meter. *Malaya J. Mat.*, 5(2):4177–4180, 2020.

[39] Taimur Shahzad Gill, Durr E Shehwar, Hira Memon, Sobia Khanam, Ali Ahmed, Urooj Shaukat, Abdul Mateen, and Syed Sajjad Haider Zaidi. IoT based smart power quality monitoring and electricity theft detection system. In *2021 16th International Conference on Emerging Technologies (ICET)*, pages 1–4. IEEE, 2021.

[40] Qi Wang, Wei Tai, Yi Tang, Ming Ni, and Shi You. A two-layer game theoretical attack-defense model for a false data injection attack against power systems. *International Journal of Electrical Power & Energy Systems*, 104:169–177, 2019.

[41] Shih-Che Huang, Yuan-Liang Lo, and Chan-Nan Lu. Non-technical loss detection using state estimation and analysis of variance. *IEEE Transactions on Power Systems*, 28(3):2959–2966, 2013.

[42] Chun-Lien Su, Wei-Hung Lee, and Chao-Kai Wen. Electricity theft detection in low voltage networks with smart meters using state estimation. In *2016 IEEE International Conference on Industrial Technology (ICIT)*, pages 493–498. IEEE, 2016.

[43] Muhammad Tariq and H. Vincent Poor. Electricity theft detection and localization in grid-tied microgrids. *IEEE Transactions on Smart Grid*, 9(3): 1920–1929, 2018.

[44] Sergio Salinas, Ming Li, and Pan Li. Privacy-preserving energy theft detection in smart grids: A P2P computing approach. *IEEE Journal on Selected Areas in Communications*, 31(9):257–267, 2013.

[45] Sergio A Salinas and Pan Li. Privacy-preserving energy theft detection in microgrids: A state estimation approach. *IEEE Transactions on Power Systems*, 31(2):883–894, 2015.

[46] Paria Jokar, Nasim Arianpoo, and Victor CM Leung. Electricity theft detection in AMI using customers' consumption patterns. *IEEE Transactions on Smart Grid*, 7(1):216–226, 2015.

[47] Mi Wen, Donghuan Yao, Beibei Li, and Rongxing Lu. State estimation based energy theft detection scheme with privacy preservation in smart grid. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[48] Eklas Hossain, Imtiaj Khan, Fuad Un-Noor, Sarder Shazali Sikander, and Samiul Haque Sunny. Application of big data and machine learning in smart grid, and associated security concerns: A review. *IEEE Access*, 7:13960–13988, 2019.

[49] Sravan Kumar Gunturi and Dipu Sarkar. Ensemble machine learning models for the detection of energy theft. *Electric Power Systems Research*, 192:106904, 2021.

[50] Madalina Mihaela Buzau, Javier Tejedor-Aguilera, Pedro Cruz-Romero, and Antonio Gómez-Expósito. Detection of non-technical losses using smart meter data and supervised learning. *IEEE Transactions on Smart Grid*, 10(3): 2661–2670, 2019.

[51] Zhongzong Yan and He Wen. Electricity theft detection base on extreme gradient boosting in AMI. *IEEE Transactions on Instrumentation and Measurement*, 70:1–9, 2021.

[52] Rajiv Punmiya and Sangho Choe. Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. *IEEE Transactions on Smart Grid*, 10(2):2326–2329, 2019.

[53] Zibin Zheng, Yatao Yang, Xiangdong Niu, Hong-Ning Dai, and Yuren Zhou. Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. *IEEE Transactions on Industrial Informatics*, 14(4): 1606–1615, April 2018.

[54] Md Nazmul Hasan, Rafia Nishat Toma, Abdullah-Al Nahid, MM Manjurul Islam, and Jong-Myon Kim. Electricity theft detection in smart grid systems: A CNN-LSTM based approach. *Energies*, 12(17):3310, 2019.

[55] Tianyu Hu, Qinglai Guo, Xinwei Shen, Hongbin Sun, Rongli Wu, and Haoning Xi. Utilizing unlabeled data to detect electricity fraud in AMI: A semisupervised deep learning approach. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3287–3299, 2019.

[56] Marcelo Zanetti, Edgard Jamhour, Marcelo Pellenz, Manoel Penna, Voldi Zambenedetti, and Ivan Chueiri. A tunable fraud detection system for advanced metering infrastructure using short-lived patterns. *IEEE Transactions on Smart Grid*, 10(1):830–840, 2017.

[57] Kedi Zheng, Yi Wang, Qixin Chen, and Yuanpeng Li. Electricity theft detecting based on density-clustering method. In *2017 IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia)*, pages 1–6. IEEE, 2017.

[58] Sandeep Kumar Singh, Ranjan Bose, and Anupam Joshi. PCA based electricity theft detection in advanced metering infrastructure. In *2017 7th International Conference on Power Systems (ICPS)*, pages 441–445. IEEE, 2017.

[59] Mohammad Al-Rubaie and Jien Morris Chang. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17(2):49–58, 2019.

[60] Qi Jia, Linke Guo, Yuguang Fang, and Guirong Wang. Efficient privacy-preserving machine learning in hierarchical distributed system. *IEEE Transactions on Network Science and Engineering*, 6(4):599–612, 2018.

[61] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

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

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

[64] Christopher Richardson, Nicholas Race, and Paul Smith. A privacy preserving approach to energy theft detection in smart grids. In *2016 IEEE International Smart Cities Conference (ISC2)*, pages 1–4. IEEE, 2016.

[65] Donghuan Yao, Mi Wen, Xiaohui Liang, Zipeng Fu, Kai Zhang, and Baojia Yang. Energy theft detection with energy privacy preservation in the smart grid. *IEEE Internet of Things Journal*, 6(5):7659–7669, 2019.

[66] Mahmoud Nabil, Muhammad Ismail, Mohamed MEA Mahmoud, Waleed Alasmary, and Erchin Serpedin. PPETD: Privacy-preserving electricity theft detection scheme with load monitoring and billing for AMI networks. *IEEE Access*, 7:96334–96348, 2019.

[67] Mohamed I Ibrahem, Mahmoud Nabil, Mostafa M Fouda, Mohamed MEA Mahmoud, Waleed Alasmary, and Fawaz Alsolami. Efficient privacy-preserving electricity theft detection with dynamic billing and load monitoring for AMI networks. *IEEE Internet of Things Journal*, 8(2):1243–1258, 2020.

[68] Mi Wen, Rong Xie, Kejie Lu, Liangliang Wang, and Kai Zhang. FedDetect: A novel privacy-preserving federated learning framework for energy theft detection in smart grid. *IEEE Internet of Things Journal*, 9(8):6069–6080, 2022.

[69] Muhammad Mansoor Ashraf, Muhammad Waqas, Ghulam Abbas, Thar Baker, Ziaul Haq Abbas, and Hisham Alasmary. FedDP: A privacy-protecting theft

detection scheme in smart grids using federated learning. *Energies*, 15(17):6241, 2022.

[70] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[71] Qi Xia, Winson Ye, Zeyi Tao, Jindi Wu, and Qun Li. A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing*, 1(1):100008, 2021.

[72] Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. CAFE: Catastrophic data leakage in vertical federated learning. *Advances in Neural Information Processing Systems*, 34:994–1006, 2021.

[73] Ajit Muzumdar, Chirag Modi, and C Vyjayanthi. Designing a blockchain-enabled privacy-preserving energy theft detection system for smart grid neighborhood area network. *Electric Power Systems Research*, 207:107884, 2022.

[74] George M Messinis, Alexandros E Rigas, and Nikos D Hatziargyriou. A hybrid method for non-technical loss detection in smart distribution grids. *IEEE Transactions on Smart Grid*, 10(6):6080–6091, 2019.

[75] Matheus Alberto de Souza, Jose LR Pereira, Guilherme de O Alves, Braulio C de Oliveira, Igor D Melo, and Paulo AN Garcia. Detection and identification of energy theft in advanced metering infrastructures. *Electric Power Systems Research*, 182:106258, 2020.

[76] Isabel Wagner and David Eckhoff. Technical privacy metrics: a systematic survey. *ACM Computing Surveys (CSUR)*, 51(3):1–38, 2018.

[77] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. In *35th Conference on Neural Information Processing Systems, NeurIPS 2021*, pages 7232–7241. Neural information processing systems foundation, 2021.

[78] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium, USENIX Security 2021*, pages 2615–2632. USENIX Association, 2021.

[79] Xinben Gao and Lan Zhang. PCAT: Functionality and data stealing from split learning by pseudo-client attack. In *USENIX Security 2023*. USENIX, 2023.

[80] Commission for Energy Regulation (CER). CER smart metering project - electricity customer behaviour trial, 2009-2010 [dataset], 2012. URL https://www.ucd.ie/issda/data/commissionforenergyregulationcer/.

[81] Muhammad Adil, Nadeem Javaid, Umar Qasim, Ibrar Ullah, Muhammad Shafiq, and Jin-Ghoo Choi. LSTM and bat-based rusboost approach for electricity theft detection. *Applied Sciences*, 10(12):4378, 2020.

[82] Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cecchet, Prashant Shenoy, and Jeannie Albrecht. Smart*: An open data set and tools for enabling research in sustainable homes. *SustKDD, August*, 111(112):108, 2012.

[83] James R. Schofield, Richard Carmichael, Simon H. Tindemans, Mark Bilton, Matt Woolf, and Goran Strbac. Low carbon london project: Data from the dynamic time-of-use electricity pricing trial, 2013 [dataset], 2015. URL https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households.

[84] Christopher Laughman, Kwangduk Lee, Robert Cox, Steven Shaw, Steven Leeb, Les Norford, and Peter Armstrong. Power signature analysis. *IEEE Power and Energy Magazine*, 1(2):56–63, 2003.

[85] U.S. Department of Energy. GridLAB-D: The next-generation simulation software, 2019. URL https://www.gridlabd.org/.

[86] Manfred Pöchacker, Tamer Khatib, and Wilfried Elmenreich. RAPSim - microgrid simulator, 2014. URL https://sourceforge.net/projects/rapsim/.

[87] Electric Power Research Institute. OpenDSS, 2020. URL https://www.epri.com/pages/sa/opendss.

[88] Edris Pouresmaeil, Juan Miguel Gonzalez, Claudio A. Canizares, and Kankar Bhattacharya. Smart Residential Load Simulator (SRLS), 2018. URL https://uwaterloo.ca/power-energy-systems-group/downloads/smart-residential-load-simulator-srls.

[89] Noah Pflugradt. LoadProfileGenerator, 2020. URL https://www.loadprofilegenerator.de/.

[90] U.S. Department of Energy. EnergyPlus, 2020. URL https://energyplus.net/.

[91] Endesa. Energy transition: The time has come for the prosumer, January 2022. URL https://www.endesa.com/en/the-e-face/energy-sector/prosumer-key-role-energy-transition.

[92] Muhammad Rizwan Asghar, György Dán, Daniele Miorandi, and Imrich Chlamtac. Smart meter data privacy: A survey. *IEEE Communications Surveys & Tutorials*, 19(4):2820–2835, 2017.

[93] Ian Clover. UK could be home to 24 million clean energy prosumers by 2050. *PV Magazine*, September 2016. URL https://www.pv-magazine.com/2016/09/27/uk-could-be-home-to-24-million-clean-energy-prosumers-by-2050-says-report_100026268.

[94] Eunice Espe, Vidyasagar Potdar, and Elizabeth Chang. Prosumer communities and relationships in smart grids: A literature review, evolution and future directions. *Energies*, 11(10):2528, 2018.

[95] Milena Radenkovic and Adam Walker. Contextual dishonest behaviour detection for cognitive adaptive charging in dynamic smart micro-grids. In *2019 15th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 44–51. IEEE, 2019.

[96] KSK Weranga, Sisil Kumarawadu, and DP Chandima. *Smart metering design and applications*. Springer, 2014.

[97] Arwa Alromih, John A Clark, and Prosanta Gope. Electricity theft detection in the presence of prosumers using a cluster-based multi-feature detection model. In *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 339–345. IEEE, 2021.

[98] Abdulrahman Takiddin, Muhammad Ismail, Usman Zafar, and Erchin Serpedin. Robust electricity theft detection against data poisoning attacks in smart grids. *IEEE Transactions on Smart Grid*, 2020.

[99] Yanlin Peng, Yining Yang, Yuejie Xu, Yang Xue, Runan Song, Jinping Kang, and Haisen Zhao. Electricity theft detection in ami based on clustering and local outlier factor. *IEEE Access*, 9:107250–107259, 2021.

[100] European Commission and Eurostat. *Manual for statistics on energy consumption in households*. Publications Office, 2013. URL https://ec.europa.eu/eurostat/documents/3859598/5935825/KS-GQ-13-003-EN.PDF/baa96509-3f4b-4c7a-94dd-feb1a31c7291.

[101] Alexander Dokumentov and Rob J Hyndman. STR: a seasonal-trend decomposition procedure based on regression. *arXiv preprint arXiv:2009.05894*, 2020.

[102] James B Elsner and Anastasios A Tsonis. *Singular spectrum analysis: a new tool in time series analysis*. Springer Science & Business Media, 2013.

[103] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition. *Journal of official statistics*, 6(1):3–73, 1990.

[104] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[105] Kevin P Schneider, Yousu Chen, David P Chassin, Robert G Pratt, David W Engel, and Sandra E Thompson. Modern grid initiative distribution taxonomy final report. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2008.

[106] Guido Van Rossum and Fred L. Drake. *Python 3 reference manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

[107] Dimitris Bertsimas, Agni Orfanoudaki, and Holly Wiberg. Interpretable clustering: an optimization approach. *Machine Learning*, 110:89–138, 2021.

[108] Taher Al-Shehari and Rakan A Alsowail. An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. *Entropy*, 23(10):1258, 2021.

[109] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[110] Prosanta Gope and Biplab Sikdar. Lightweight and privacy-friendly spatial data aggregation for secure power supply and demand management in smart grids. *IEEE Transactions on Information Forensics and Security*, 14(6):1554–1566, 2018.

[111] Chandra Thapa, Mahawaga Arachchige Pathum Chamikara, and Seyit A Camtepe. Advancements of federated learning towards privacy preservation: from federated learning to split learning. In *Federated Learning Systems: Towards Next-Generation AI*, pages 79–109. Springer, 2021.

[112] Andrew Paverd, Andrew Martin, and Ian Brown. Modelling and automatically analysing privacy properties for honest-but-curious adversaries. *University of Oxford Technical Report*, 2014.

[113] Daniel Bernau, Jonas Robl, Philip W Grassal, Steffen Schneider, and Florian Kerschbaum. Comparing local and central differential privacy using membership inference attacks. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 22–42. Springer, 2021.

[114] Jingwen Zhao, Yunfang Chen, and Wei Zhang. Differential privacy preservation in deep learning: Challenges, opportunities and solutions. *IEEE Access*, 7: 48901–48911, 2019.

[115] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. Unleashing the tiger: Inference attacks on split learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 2113–2129, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384544.

[116] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6): 2769–2794, 2007.

[117] R. Can Aygun and Ali Gokhan Yavuz. Network anomaly detection with stochastically improved autoencoder based models. In *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pages 193–198. IEEE, 2017.

[118] Hasan Torabi, Seyedeh Leili Mirtaheri, and Sergio Greco. Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity*, 6 (1):1, 2023.

[119] Andrea Borghesi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. Anomaly detection using autoencoders in high performance computing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9428–9433, 2019.

[120] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[121] Benjamin Zi Hao Zhao, Aviral Agrawal, Catisha Coburn, Hassan Jameel Asghar, Raghav Bhaskar, Mohamed Ali Kaafar, Darren Webb, and Peter Dickinson. On the (in)feasibility of attribute inference attacks on machine learning models. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 232–251. IEEE, 2021.

[122] Shagufta Mehnaz, Sayanton V. Dibbo, Ehsanul Kabir, Ninghui Li, and Elisa Bertino. Are your sensitive attributes private? novel model inversion attribute inference attacks on classification models. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4579–4596, 2022.

[123] Jinyuan Jia and Neil Zhenqiang Gong. Defending against machine learning based inference attacks via adversarial examples: Opportunities and challenges. *Adaptive Autonomous Secure Cyber Systems*, pages 23–40, 2020.

[124] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[125] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[126] Alaa Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 17(1):168–192, 2020.

[127] Sharif Abuadbba, Kyuyeon Kim, Minki Kim, Chandra Thapa, Seyit A Camtepe, Yansong Gao, Hyoungshick Kim, and Surya Nepal. Can we use split learning on 1D CNN models for privacy preserving training? In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 305–318, 2020.

[128] Valeria Turina, Zongshun Zhang, Flavio Esposito, and Ibrahim Matta. Combining split and federated architectures for efficiency and privacy in deep learning. In *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, pages 562–563, 2020.

[129] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[130] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[131] Paul Bolton. Gas and electricity prices under the energy price guarantee and beyond. Technical report, Commons Library Research Briefing,

2023. URL https://researchbriefings.files.parliament.uk/documents/CBP-9714/CBP-9714.pdf.

[132] EATON. Blackout tracker - United States annual report 2017. Technical report, EATON, 2017. URL https://www.eaton.com/explore/c/us-blackout-tracker--2?x=NzOhds.

[133] Shahab Bahrami, Yu Christine Chen, and Vincent WS Wong. Deep reinforcement learning for demand response in distribution networks. *IEEE Transactions on Smart Grid*, 12(2):1496–1506, 2020.

[134] Yuanzhang Xiao and Mihaela van der Schaar. Dynamic stochastic demand response with energy storage. *IEEE Transactions on Smart Grid*, 12(6): 4813–4821, 2021.

[135] Huseyin Burak Akyol, Chris Preist, and Daniel Schien. Avoiding overconfidence in predictions of residential energy demand through identification of the persistence forecast effect. *IEEE Transactions on Smart Grid*, 2022.

[136] Muneeb Ul Hassan, Mubashir Husain Rehmani, Jia Tina Du, and Jinjun Chen. Differentially private demand side management for incentivized dynamic pricing in smart grid. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[137] Ofgem. Tackling electricity theft. Technical report, Ofgem, 2013. URL https://www.ofgem.gov.uk/sites/default/files/docs/2013/07/20130703_tackling-electricity-theft.pdf.

[138] Jie Yin and Xuefeng Yan. Stacked sparse autoencoders that preserve the local and global feature structures for fault detection. *Transactions of the Institute of Measurement and Control*, 43(16):3555–3565, 2021.

[139] Mohammad Malekzadeh, Richard G Clegg, and Hamed Haddadi. Replacement autoencoder: A privacy-preserving algorithm for sensory data analysis. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 165–176. IEEE, 2018.

[140] Baptiste Wicht. *Deep learning feature extraction for image processing*. PhD thesis, University of Fribourg, 2017.

[141] Wes McKinney. Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[142] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[143] Diego Peteiro-Barral and Bertha Guijarro-Berdiñas. A survey of methods for distributed machine learning. *Progress in Artificial Intelligence*, 2(1):1–11, 2013.

Appendices

Appendix A

Neural Networks

Neural networks (NN), like any other network or graph, are networks that are composed of nodes *(neurons)* and edges *(weights)*. The nodes or neurons are arranged into layers starting from the input layer, followed by one or more hidden layers and finally the output layer. Each neuron is a computational unit that takes the inputs from the preceding layer and outputs the weighted sum of these inputs (plus a bias). The output of each neuron can be restricted using *activation functions*. The most used activation functions are *Rectified Linear Unit (ReLu)*, *hyperbolic Tangent function (Tanh)* and *Sigmoid*. These activation functions limit the output value within a specified range, i.e., ReLu output is from 0 to +infinity, Tanh output is from -1 to 1 and Sigmoid output ranges between 0 and 1. Therefore, each neuron n in layer l calculates its output z_n^l as:

$$z_n^l = A_z^l(\sum_{j=1}^{s}(i_j^{l-1} \times w_j^l) + b_n) \tag{A.1}$$

where A is the activation function, i_j^{l-1} is the jth output from the preceding layer, w_j^l is the jth weight of that output and b_n is the bias of this neuron. There are two passes in each round (epoch) of training a neural network: a forward propagation pass and a backward propagation pass (backpropagation). In the forward pass, the input data are propagated to the input layer, then proceed to the hidden layer(s), measuring the network's predictions up to the output layer where the network outputs the prediction $\hat{y}$. This makes $\hat{y}$ equals to:

$$\hat{y} = A^L(W^L A^{L-1}(W^{L-1}....A^2(W^2 A^1(W^1 X))...)) \tag{A.2}$$

where L is the total number of layers, W^i is the weights vector of layer i and X is the input vector. This is first done using initial weights and bias (weights and

bias are initialised randomly). The outputs of all neurons of the same layer are called activations. The network's error (loss) is calculated based on the output of the forward pass prediction $\hat{y}$ and the desired output y. The loss function is computed for every output of the neural network as follows: $loss = L(\hat{y}, y)$. In the backpropagation pass, the weights and biases of the network are adjusted in proportion to how much they contribute to the overall error (loss). These adjustment values are called gradients and they are sent back to along the network to update the neurons weights and bias where the updated value for each weight w will be: $w_{new} = w_{old} - \alpha(\frac{\partial loss}{\partial w})$, where α is a learning rate that controls how much we are adjusting the weights with respect the loss gradient and ∂ is the derivative of the loss in respect to the that weight w.

Appendix B

Split Learning Algorithm

This appendix provides a brief description of the three main functions in the Split Learning algorithm.

Algorithm 2 Split learning algorithm

---

function SERVER ▷ executes at round $t \geq 0$
  for client $c \in S_t$ **do**
    $A_{c,t} \leftarrow \text{CLIENTUPDATE}(c, t)$
    Complete forward propagation with $A_{c,t}$ to get $A_{S,t}$
    Calculate Loss
    $W_{S,t+1} \leftarrow W_{S,t} - \eta \nabla\, l(W_{S,t}; A_{S,t})$ ▷ Back propagation part for the server
    $\text{CLIENTBACKPROP}(c, t, \nabla \ell(A_{S,t}; W_{S,t})))$
  end for
end function

function $\text{CLIENTUPDATE}(c, t)$
  $A_{c,t} \leftarrow \phi$
  if Client c is first client in $t = 0$ **then**
    $W_{c,t} \leftarrow randominitialize$
  else
    $W_{c,t} \leftarrow \text{CLIENTBACKPROP}(W_{c-1,t-1})$
  end if
  for local epoch e **do**
    for batch $b \in B$ **do**
      Forward propagation on the client part
      Concatenate the activations of cut layer to $A_{c,t}$
    end for

end for
send $A_{c,t}$ to Server
end function

function ClientBackprop($c, t, \nabla\ell(A_{S,t}; W_{S,t})$)
for batch $b \in B$ **do**
Backpropagation on client part with $\eta\nabla(A_{S,t}; W_{S,t})$
end for
Update model weights $W_{c,t+1}$ and send to next client
end function