UNIVERSITY OF THE PHILIPPINES

Bachelor of Science in Electrical Engineering

Bachelor of Science in Electronics and Communications Engineering

Carl Samuel R. Lavilla

Michael E. Osorio

Zildjian Joshua C. Restituto

Effect of Net Metering and Rooftop Photovoltaics on Electricity Theft Detection

Undergraduate Project Adviser:

Adonis Tio, M.S. EE

Electrical and Electronics Engineering Institute

University of the Philippines Diliman

Undergraduate Project Reader:

Nicolette Arriola, M.S. EE

Electrical and Electronics Engineering Institute

University of the Philippines - Diliman

Date of Submission

July 19, 2021

Permission is given for the following people to have access to this thesis:

Circle one or more concerns: I P C	
Available to the general public	Yes/No
Available only after consultation with author/thesis adviser	Yes/No
Available only to those bound by confidentiality agreement	Yes/No

Students' signatures:

Signature of undergraduate project advisers:

University Permission Page

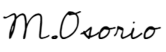
I hereby grant the University of the Philippines non-exclusive worldwide, royalty-free license to reproduce, publish, and public distribute copies of this work in any form subject to the provisions of applicable laws, the provisions of the UP IPR policy and any contractual obligations, as well as more specific permission marking on the Title Page.

Specifically I grant the following rights to the University:

- to upload a copy of the work in the theses database of the college/school/institute/department and in any other databases available on the public internet;
- to publish the work in the college/school/institute/department journal, both in print and electronic or digital format and online; and
- to give open access to above-mentioned work, thus allowing "fair-use" of the work in accordance with the provisions of the Intellectual Property Code of the Philippines (Republic Act No. 8293), especially for teaching, scholarly, and research purposes.

Carl Samuel R. Lavilla — 07/18/21

Carl Samuel R. Lavilla — Date



Michael E. Osorio — 07/18/21

Michael E. Osorio — Date

Zildjian Joshua C. Restituto — 07/18/21

Zildjian Joshua C. Restituto — Date

Approval Sheet

In partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical Engineering, and Bachelor of Science in Electronics and Communications Engineering, this project entitled "Effect of Net Metering and Rooftop Photovoltaics on Electricity Theft Detection", prepared and submitted by Carl Samuel R. Lavilla, Michael E. Osorio, and Zildjian Joshua C. Restituto, is hereby recommended for approval.

Adonis Tio, M.S. EE Adviser	Date

Accepted in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical Engineering, and Bachelor of Science in Electronics and Communications Engineering.

Nicolette Arriola, M.S. EE Panel Member	Date
Dr. Michael Angelo Pedrasa Director, Electrical and Electronics Engineering Institute	Date

Effect of Net Metering and Rooftop Photovoltaics on Electricity Theft Detection

Undergraduate Student Project

by

Carl Samuel R. Lavilla
2015-05369
B.S. Electrical Engineering

Michael E. Osorio
2015-05350
B.S. Electronics and Communications Engineering

Zildjian Joshua C. Restituto
2015-00863
B.S. Electrical Engineering

Adviser:

Adonis Tio, Ph.D.

University of the Philippines, Diliman

July 2021

Abstract

Effect of Net Metering and Rooftop Photovoltaics on Electricity Theft Detection

This paper presents the effects of rooftop photovoltaics and Net Metering on existing methods of theft detection algorithms. Three algorithms were tested: (a) Support Vector Machine, (b) Artificial Neural Network, (c) Anomaly Coefficient Calculation. The algorithms were tested on different datasets with varying presence of PV and Net Metering. The datasets were created using Python and OpenDSS. The accuracy, precision, detection rate and F1 score of the algorithms were documented and analyzed. For SVM and ANN, there is a decreasing trend in the metric scores as the penetration levels of PV and Net Metering is increased. The Anomaly Coefficient Calculation is consistent all throughout the different cases, but requires a sufficient amount of data points to correctly detect and identify pilferage.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Industrial, commercial and residential buildings consume electricity on a day to day basis. Power distribution companies use electricity meters to calculate the electricity consumption of their customers. The meter calculation is shown in our electricity bills, together with the breakdown of the total electricity charges. These calculations are often affected by technical and non-technical losses in the system. To adapt to the continuous evolution of the power industry and its electrical systems, power distribution companies would also need to advance their system loss reduction technology.

1.1 Rooftop Photovoltaics and Net Metering

The rising cost of fossil fuel and its adverse effect on the environment have driven the search for a more sustainable way of producing power. The Philippines, as one of the emerging countries in Asia, has seen steadily increasing electricity demand in recent years. In 2019, the total electricity consumption grew by 6.3% and reached a total peak demand of 15581 MW [1]. Although the country's energy generation is mainly reliant on fossil fuels, renewable energy generation through solar photovoltaics (PV) has significantly increased over the past few years. Installed solar capacity increased from 23 MW in 2014 to 921 MW in 2019 [2]. Through the years, the cost of installing PV systems has been continuously decreasing, making it an extremely popular means of producing cost-effective power [3].

The use of rooftop PV in residential and industrial facilities not only addresses the problems of increasing demand and environmental pressure but can also save expenses in electrical bills via Net Metering. Under the Renewable Energy Act of 2008 [4], Net Metering allows distribution grid users to sell excess en-

ergy back to the grid. By utilizing a bi-directional meter called the *net meter*, a prosumer is billed every month only for the net electricity consumed [5].

Figure 1.1 illustrates how net metering works. The numbers in the figure represent the following: (1) solar PV installation, (2) DC-AC converter, (3) home electrical panel, (4) bi-directional meter, (5) electrical grid. Net metering allows users to sell their excess energy back to the grid at market rates. For example, if your PV generates 900 kwH of power in a month, and your household only uses 500 kwH from the electrical grid, then the power distribution company owes you will pay you for the excess 400 kWh of power.

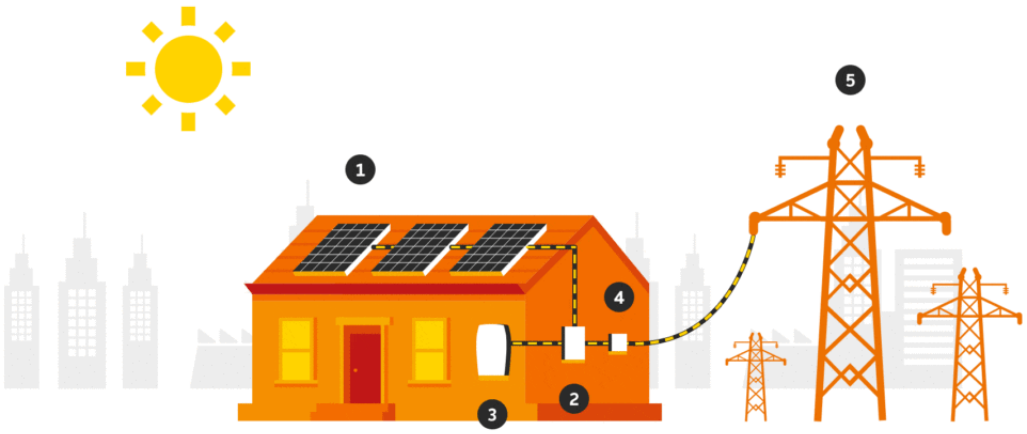


Image from [6]

Figure 1.1: How Net Metering Works

1.2 Types Of Losses

Electricity access plays an important role in the development of the society. The population grows in hunger for more power to supply their needs, posing more and more challenges for the energy sector. One of the key factors that affect the efficiency of power distribution is system loss, which refers to the energy that is not received by the consumer. As illustrated in Figure 1.2, there are two types of losses: technical and non-technical losses. Technical loss is inherent due to the physical delivery of electricity through distribution lines and other substation equipment. These are usually taken into account and are foreseen by the distributors. Non-technical losses (NTL) are results of unidentified energy flows, usually from illegal connection, tampering of meters and erroneous reading [7].

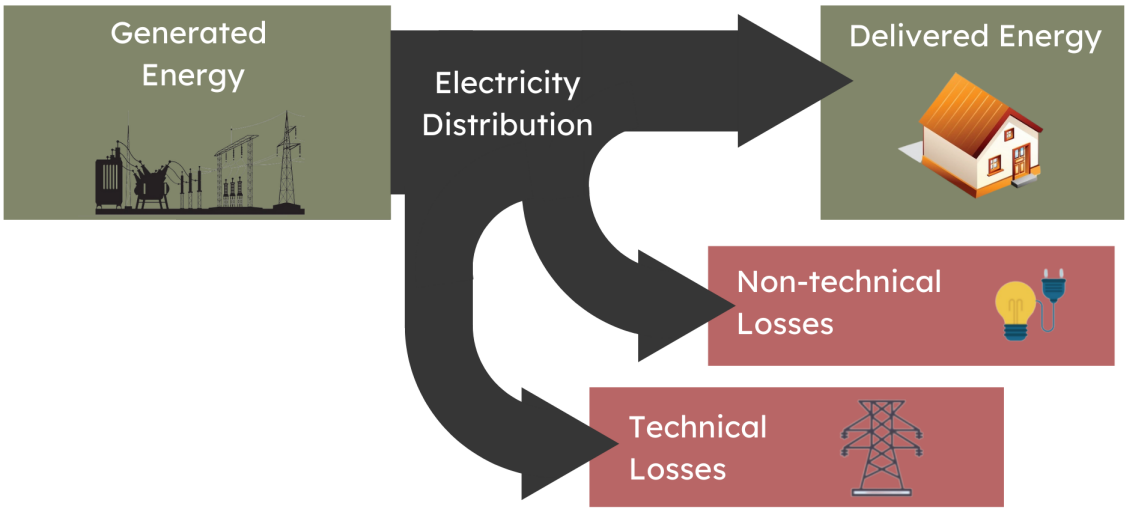


Figure 1.2: System Loss

1.3 Electricity Theft

Electricity theft is caused by pilferers that perform illegal alterations on the system in order to reduce their own power consumption reading. In some cases, it results to an increase in the electricity consumption recorded by the smart meter of other users [8]. Non-technical loss in smart grids that are mostly due to energy theft, has become a major enduring challenge in the power distribution industry worldwide [9]. For the electricity providers, electricity theft is a big nuisance and they have been trying to detect such activities by deploying smart meters (SMs) to modernize current power grids and energy metering functions. However, these SMs are very vulnerable to various types of attacks and may make energy theft easier to commit [9]. For the consumer, a discrepancy in meter reading might mean paying for more than electricity consumed.

Due to the nature of non-technical loss during transmission of electrical energy, it is difficult for the utility companies to detect and apprehend the people responsible for theft. Various works claim that energy theft in advanced metering infrastructure (AMI) is becoming more prevalent [10]. With this, the energy sector is challenged to call for the development of effective theft detection techniques.

1.4 Project Flow and Organization

Chapter 2 discusses the related works about system loss reduction, electricity theft, and some detection electricity theft methods. A section will also be given to understanding related works on rooftop photovoltaics in theft detection.

Chapter 3 includes the statement of the problem that lead to the idea of the project and the objectives together with its scope and limitations.

Chapter 4 contains the methodology explaining how the project is implemented. It consists of the overview, modelling of the test system, implementation of theft detection algorithms, and the performance metrics calculation.

Chapter 5 compares the results and discusses how PV and Net Metering affects the different algorithms used.

Chapter 6 contains the conclusion and recommendation that the proponents have for this project.

Chapter 2

Background and Related Work

In this chapter, different types of electricity theft as well as works related to electricity theft detection algorithms are discussed and analyzed. The concept of system loss is also expanded in this chapter as well as the current state of electricity theft detection algorithms with regards to systems with rooftop photovoltaics.

2.1 System Loss Reduction

System loss is present in all electrical systems. Electrical equipment and materials, such as transmission lines and transformers that are used in distribution infrastructures, dissipate heat and energy is wasted in the process. Figure 2.1 shows a model for a short transmission line. Transmission lines are classified according to their lengths; with the short line as the most basic model. When power is transmitted along the line, a difference in power is caused by the series impedance of the line. The power loss is denoted as $P_{losses} = I_s^2(R + j\omega L)$ [11].

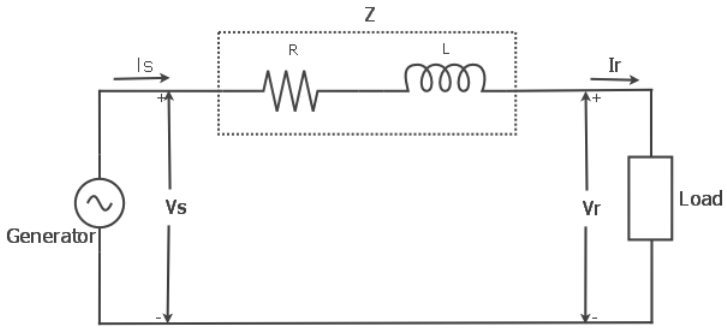


Figure 2.1: Transmission Lines - Short Line Model

System loss significantly affects the revenue of electric utilities, and even its customers. This motivates growing countries to develop different methods and initiatives to reduce system loss, involving the government and the general public in the process. In the Philippines, the Manila Electric Company (Meralco) implements reduction initiatives that reduce both technical and non-technical losses. [12]

For technical loss reduction, re-configuring and upgrading distribution systems are regularly done to improve the overall efficiency of electric power distribution across the country. This includes the installation of capacitor banks and optimizing the loading of distribution transformers.

For non-technical loss reduction, the Anti-Pilferage Law (Renewable Energy Act of 1998) [13] is strictly implemented to apprehend offenders and prevent electricity theft. Despite this, different types of electricity theft are now prevalent, even in advanced electrical systems like AMIs. To combat this, our current theft detection methods will need to adapt.

2.2 Electricity Theft

There are various types or methods of electricity theft. These methods can range from direct hooking from line to cyber-attacks on smart meters. However, physical methods of theft are no longer the only type of attack. With the increasing number of smart meters, cyber-attacks are on the rise [14][15]. Electricity theft could be divided into three groups, namely, interruption of measurement, tampering of stored demand, and network hijacking as follows [10]:

1. Interruption of measurement is done by the attacker by either disconnecting the meter or via meter inversion. This takes place before the meter could make a record of the consumed electricity.

2. Tampering of stored demand occurs when the attacker accesses the administrative interfaces with passwords and erases the stored value of the measured electricity consumption.

3. Network hijacking is completed when the attacker is able to intercept communication and inject traffic to the smart meter allowing a delivery of forged data to the smart meter.

Clients of electric companies make attempts at stealing energy provided by their very own electric companies, with varying degrees of success. The most common reason for this phenomenon is the attempt to save money. Among the many known methods of theft, the most popular ones are: making a concealed connection and a mechanical blockade of the analog mechanism. Energy theft by tampering with a digital smart meter is much more difficult and requires specialized knowledge, which greatly narrows the circle of clients whose specialized knowledge enables them to perform unauthorized manipulation of the meter [16].

2.3 Electricity Theft Detection Methods

Electricity theft detection methods aim to distinguish pilferers in an electrical system. Theft detection methods that are used in smart grids include the use of a check meter. A check meter is installed upstream as basis, which compares its reading to the sum of individual smart meter readings of downstream households. The check meter's role is to check whether or not there is a mismatch between the total reading and the individual household readings.

To illustrate how theft detection works, an example of a smart grid is shown in Figure 2.2. In this case, $M3$ commits a cyber-attack which manipulates their own smart meter reading from 40 kWh to 20 kWh. However, this does not affect the reading of the check meter; instead, the check meter still reads 40 kWh for $M3$ as shown in Equation 2.1.

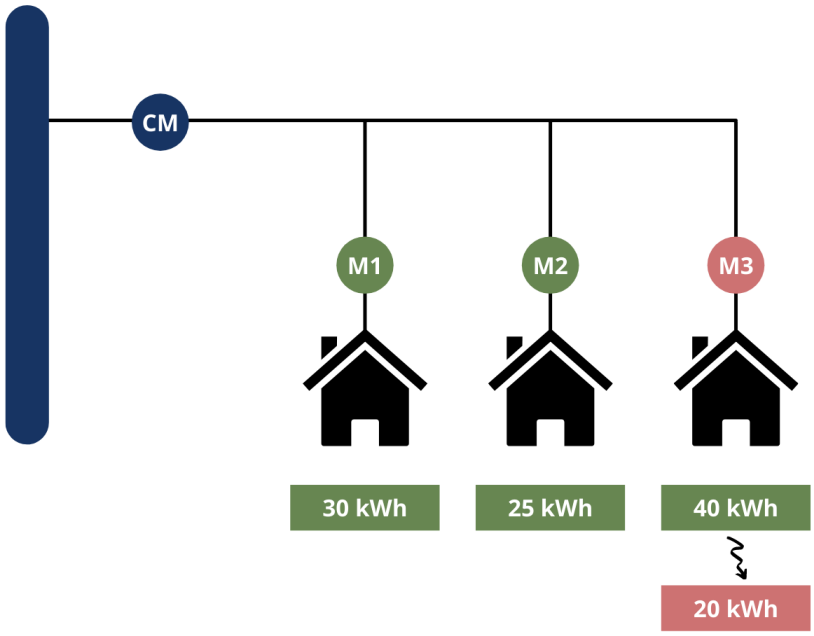


Figure 2.2: Smart Grid Model - M3 as Pilferer

$$\begin{aligned} CM &= M1 + M2 + M3 + P_{techlosses} \\ CM &= 30kWh + 25kWh + 40kWh + P_{techlosses} \\ CM &= 95kWh + P_{techlosses} \end{aligned} \tag{2.1}$$

The manipulated value of $M3$ is taken into account when calculating for the sum of the individual smart meter readings in the grid. As seen in Equation 2.2, the sum of the individual readings is not equal to the check meter reading which is 95 kWh. The obvious mismatch means that there is pilferage in the system.

$$\begin{aligned} SUM(M1, M2, M3) &= 30kWh + 25kWh + 20kWh \\ SUM(M1, M2, M3) &= 75kWh \end{aligned} \tag{2.2}$$

However, it is important to note that $P_{techlosses}$ is also relevant in the equation. In real life situations, $P_{techlosses}$ is not equal to zero. Therefore, even if there is no pilferage in the system, there would still be a

mismatch between the check meter reading and the sum of all individual meter readings. This complicates electricity theft detection especially in large systems.

Electricity theft detection methods have also evolved to adopt several machine learning techniques and use it to detect pilferage in a system. A research by [17] used Support Vector Machines (SVM) to preselect suspicious customers using historical customer consumption data. The suspected customers are then visited on-site for inspections. In this method, historical data must be collected and used to train and test the SVM model. SVM classifiers are used for binary classification problems using data with either linear or non-linear relationships. Labelled training data from two classes are mapped as points in space and a separating hyperplane is solved to separate the two classes with maximum margin. Testing data is then mapped into the same space and outputs a predicted class based on the same hyperplane [18].

Theft detection methods based on artificial neural networks (ANN) are also quickly becoming popular [19]. In a study by [20], a wide and deep convolution neural network (CNN) model is proposed as a model for theft detection in smart grids. The resulting model was able to outperform other theft detection methods such as SVM. The idea of ANN is inspired by the biological model of a neuron. A typical neural network consists of an input layer, an output layer, and several hidden layers in between [21]. Each layer is made up of multiple neurons that use the neurons of the preceding layer as its inputs and passes its output to the succeeding layer [22]. Like SVM, ANNs are trained with labelled data and is evaluated using new, unseen data.

Reference [23] proposed a method on anomaly coefficient calculation to be used as linear equations to detect pilferage. This method has been demonstrated to detect a single pilferer in a low-voltage network with household smart meters and a check meter. His succeeding works improved on this algorithm as it can now be used to detect multiple pilferers and work with small percentages of system losses. The algorithm was later tested on a larger test system with 60 households in a low-voltage network and varying theft behavior. The lowest accuracy rate was 89% and the lowest average accuracy was 97%. The highest accuracy and highest average accuracy are both 100% [12].

The algorithms that were stated above were developed and tested on systems that do not have rooftop PVs and Net Metering. The following section will discuss the technicalities and the current state of electricity theft detection in systems with PV and Net Metering.

2.4 Electricity Theft Detection in Systems with PV and Net Metering

Increasing levels of rooftop PV penetration (defined as the ratio of the total number of households with PV installation to the total number of households) introduces technical challenges that can affect the distribution grid. Related works show that factors such as voltage and current imbalance, and power losses received the most attention in recent publications [24]. As discussed in 1.1, net metering allows excess energy produced by rooftop PVs to flow back to the grid. This can also cause complications to the distribution grid because net metering introduces the concept of 'negative' power flow in the system.

As the aforementioned complexities are added to electrical systems with PV and Net Metering, current electricity theft detection methods may also be affected. To further explain this, it is important to know how theft detection algorithms work when subject to different penetration levels of PV and Net Metering. Different cases wherein pilferers manipulate their own PV generation also pose a threat to power distribution companies.

Reference [25] presented an algorithm which uses the least squares approach to detect electricity theft. Their research focused mainly on cases where pilferers increase the value of their PV generation meter reading for extra profit. Their algorithm was tested on electrical systems that use individual meters for PV generation, and check meters to detect electricity theft. Their research did not mention the use of Net Metering.

Reference [26] adopted deep machine learning techniques to detect electricity theft cyber-attacks in a PV system. Their research also developed their own dataset, including different cyber-attack functions, to realistically simulate theft. Different deep machine learning models were tested, and their metric scores were compared with each other. However, the effect of PV and Net Metering wasnot highlighted and quantitized; they focused on testing their machine learning models on datasets with PV.

2.5 Background Summary

The research done by [17] and [20] in using machine learning techniques for detecting electricity theft showed outstanding results. However, there are no PVs in the electrical systems they used. On the other hand, both works by [25] and [26] explored theft detection algorithms on test networks with PV installation. However, the focus was on the performance of the algorithms themselves and not the effect of PV and Net Metering on electricity theft detection. Therefore, there is a need to characterize the effect of PV and Net Metering on the performance of electricity theft detection to bridge the gap in knowledge.

Chapter 3

Problem Statement and Objectives

3.1 Problem Statement

The use of rooftop photovoltaics and Net Metering are on the rise. However, there have only been few discussions on the effects of such technologies with our current types of electricity theft detection algorithms. This becomes problematic when a system heavily relies on current algorithms that may not be able to detect pilferers accurately.

3.2 Objectives of the Project

This project aims to analyze the effect of rooftop photovoltaics and Net Metering on the performance of different electricity theft detection algorithms. The algorithms were tested on different datasets with varying presence of PV and Net Metering. Their accuracy, precision, recall, and F1 score was calculated and discussed.

3.3 Scope and Limitations

This project will test three theft detection algorithms: Support Vector Machine (SVM), and Artificial Neural Network (ANN), and Anomaly Coefficient Calculation. The performance of each algorithm will be tested on different datasets, each with varying presence of PV and Net Metering. The simulation will be limited to the detection of electricity theft through a cyber-attack, where the pilferer manipulates their own smart meter reading.

Chapter 4

Methodology

4.1 Overview

This chapter discusses the methodology of the project. Each section provides a comprehensive explanation of each step, from the modelling of test systems up to the implementation of theft detection algorithms.

Selected household load and PV generation data were extracted and were used as inputs to a test network model. The power-flow of the network was performed to simulate meter readings across different PV and Net Metering scenarios. Table 4.1 shows the seven dataset categories, with each of them having different PV and Net Metering penetration levels.

Dataset	**PV Penetration %**	**NM Penetration %**
D1	0	0
D2	20	0
D3	70	0
D4	20	20
D5	20	70
D6	70	20
D7	70	70

Table 4.1: Dataset Categories

Electricity theft in the form of smart meter manipulation was applied on the simulated data. The desired features were extracted, and the datasets were labeled.

Finally, these datasets were used as inputs to three different theft detection algorithms: Support Vector Machine (SVM), and Artificial Neural Network (ANN), and Anomaly Coefficient Calculation. The performance metrics of each algorithm were then calculated and assessed.

4.2 Modelling of Test Systems

4.2.1 Household Data Preparation

An Australian distribution network with household load and rooftop PV generation data was used as the primary sample data [27]. It was made available by Ausgrid, an electric utility in Sydney and nearby areas in New South Wales. The dataset contains meter readings from 300 customers taken at half-hourly intervals, and with the following parameters: PV production, generator capacity, electricity consumption, and off-peak-controlled consumption. The postcode location map and the irradiation map are shown in Figure 4.1. Irradiance is defined as the radiant energy per unit time that strikes a unit horizontal area per unit wavelength interval [28]. By observing the map, it can be inferred that the irradiation data of the households in the area covered by Ausgrid are closely correlated to each other. The data was collected from [29].

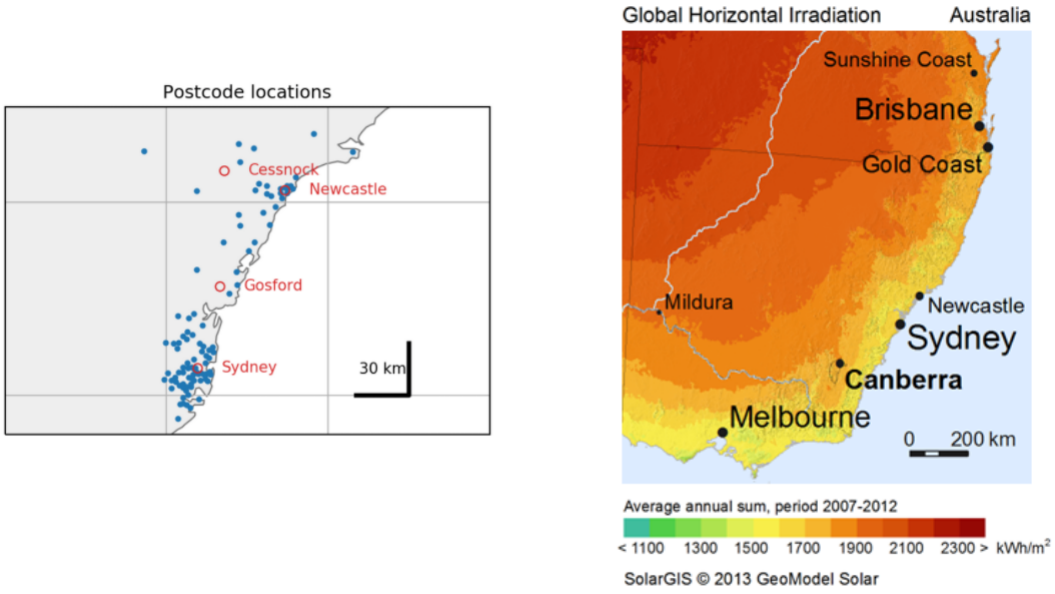


Image from [29]

Figure 4.1: Ausgrid Data Location and Irradiation Maps

Data cleaning is necessary to maintain consistency all throughout the process. The following requirements and steps were followed in preparing the household load dataset. Python was used in this process.

1. Only households with complete general consumption and gross generation data in the selected dates were selected. This reduced the number of customers of the Ausgrid data from 300 to 161.

2. One week worth of data was extracted. The week of December 5-11, 2010 was chosen for its similar temperature compared to the summer season in the Philippines.

3. The general consumption data and the gross generation data were collected and converted into individual customer profiles to be used as input load curves and PV generation curves respectively. Other irrelevant parameters, like the controlled load consumption data, were not included.

After the data cleaning process, the data is now ready to be used as input to the test network. The following section dissects the network modelling process and its specifications.

4.2.2 Network Modelling

4.2.2.1 Low-Voltage Test Feeder

The network was modeled after the IEEE European Low Voltage Test Feeder. It was designed by the Test Feeders Working Group of the Distribution System Analysis Subcommittee of the Power Systems Analysis, Computing, and Economics (PSACE) Committee [30]. The feeder is at the voltage level of 416 V (phase-to-phase). Its base frequency of 50 Hz was kept at the same level. The other parameters such as the line and substation codes were kept at a default setting.

There are 55 PQ loads that are connected to the main substation. Additionally, we placed check meters to group the households according to their locations. The topology of the system is shown in Figure 4.2. The area under each check meter is also highlighted in red. The following subsections will expound on how the household loads were modeled, depending on the presence of PV and Net Metering.

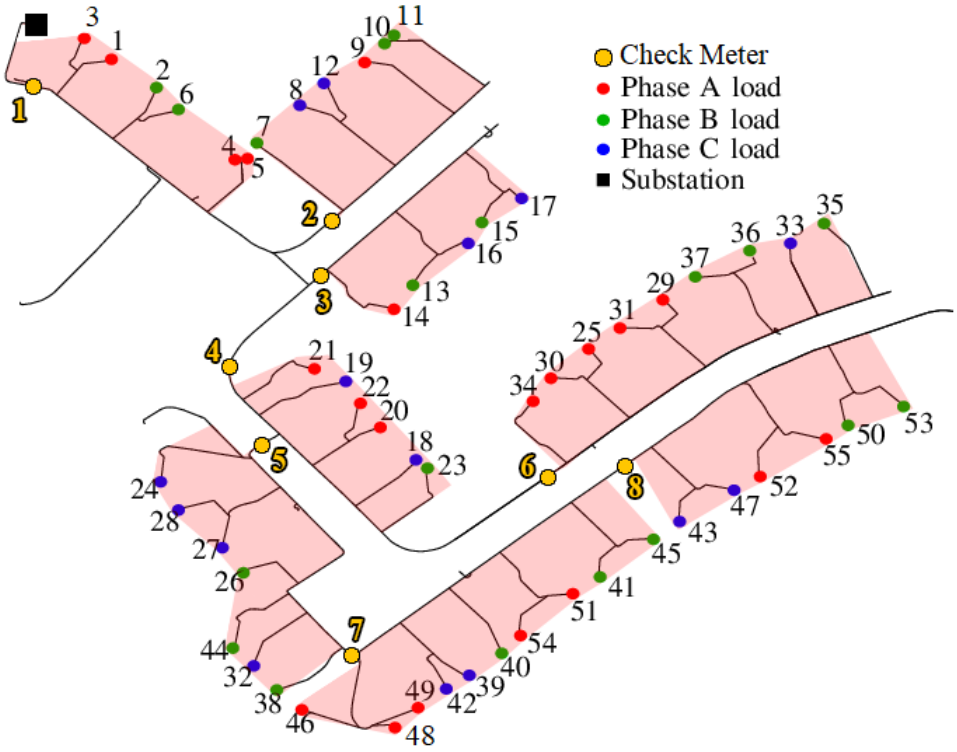


Image from [30]

Figure 4.2: System Topology

4.2.2.2 Household Loads

Each household was modelled as a PQ load, the load curve of each household was derived from the Ausgrid data. Additionally, smart meters were connected to every household to measure the power reading or consumption of each load. The sample load curve in 4.3 shows the kWh power reading of a household in 1 week at a half-hour interval.

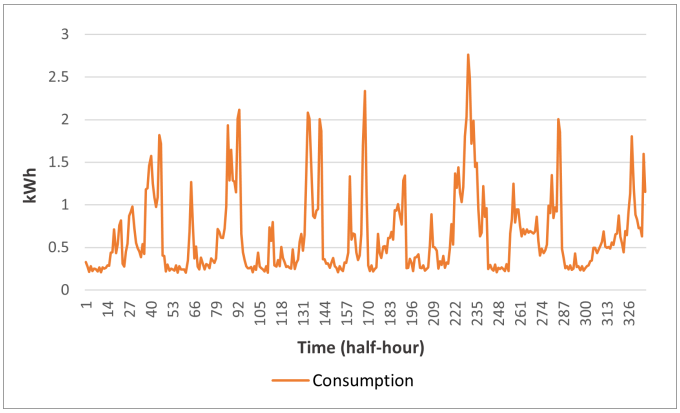


Figure 4.3: Household Load Curve - 1 Week

4.2.2.3 Rooftop PV and Net Metering Model

To simulate a household with a rooftop PV, an additional generator object was connected to the PQ load. The PV generation data from Ausgrid is used as an input for the generator. The household with a rooftop PV would have two curves: a load curve that represents its consumption, and a generation curve that represents its PV generation. For a household with Net Metering, excess energy produced by the rooftop PV will be exported to the grid. The exported energy to the grid by household i is given by

$$P_{excess,i} = P_{generation,i} - P_{consumption,i}$$

The graph in Figure 4.4 shows the kWh power reading of a household with Net Metering, together with its PV generation in 1 week at a half-hour interval. Excess energy is exported during the time-frames where PV generation is greater than the household's consumption.

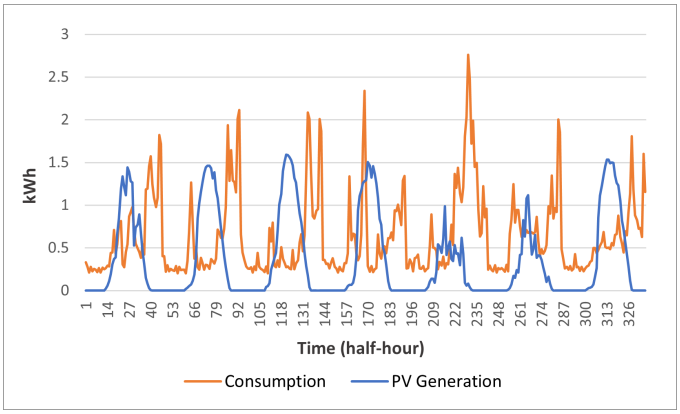


Figure 4.4: Household with PV and NM - Load and Generation Curve - 1 Week

For a household with PV but no Net Metering, excess energy is not exported back to the grid. To model this behavior, the PV generation curve was altered such that PV generation is clipped to the household consumption curve. This way, all the energy produced from the rooftop PV is consumed by the household. Figure 4.5 shows the counterpart of Figure 4.4 if the household has no Net Metering.

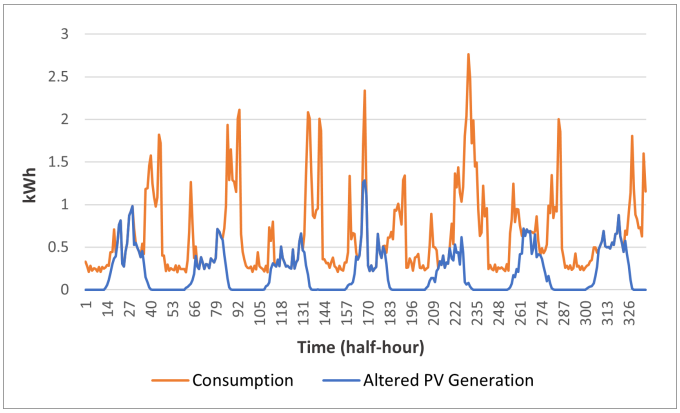


Figure 4.5: Household with PV only - Load and Generation Curve - 1 Week

After modelling the test network, it is now ready for simulation. The next section discusses the details of the power-flow simulation, and what outputs were exported from the simulations.

4.2.3 Power-flow Simulation

After modelling the test network, OpenDSS was used for power-flow simulation. It is a comprehensive power system simulation software that allows users to run power-flow calculations on a system. The

specifications of the feeder that were discussed in are required to run the power-flow simulation. Additionally, the duration of the simulation would also need to be specified. In this project specifically, OpenDSS was used to export the time-series data from the check meters and the individual household meters after the simulation.

As seen in Figure 4.2, there are 55 households that are connected to the substation, and 8 check meters were assigned based on the geographical locations of the households. To highlight the effect of Net Metering and rooftop PVs on electricity theft detection, the penetration levels of PV and Net Metering were varied. The PV penetration levels were varied between high (70% of the households have PVs), low (20% of the households have PVs) and zero (0% of the households have PVs). For the cases with PV penetration, the Net Metering penetration levels were also varied between high, low, and zero. This resulted in seven different combinations, or seven dataset categories as seen in Table 4.1.

For example, at 70% PV and 20% Net Metering, the number of households with PV is $\lfloor 55 \times 0.7 \rfloor = 38$ and the number of households with both PV and Net Metering is $\lfloor 38 \times 0.2 \rfloor = 7$. Net Metering can only be installed on households with rooftop PV.

For each of the seven datasets, 110 week-long simulations were done to serve as benign data or data without theft. Subsequently, the power readings of the check meters and household meters were exported. The reasoning behind the number of simulations is explained in Section 4.2.4.

4.2.4 Electricity Theft Representation

After the power-flow simulation, each week-long simulation was replicated to add electricity theft. The theft was represented as a cyber-attack, where the pilferer manipulates their own smart meter reading. For each week-long simulation, only one household was chosen to be a pilferer.

The theft was represented as a multiplier k_{et} to a household's meter reading. k_{et} was randomly generated using a Gaussian Distribution curve with a mean of 0.5 and standard deviation of 0.05. The minimum and maximum values of k_{et} are 0.35 and 0.65 respectively; both values are 3 standard deviations away from the mean. k_{et} is unique for every week-long simulation from Section 4.2.3.

The theft duration or frequency was varied to be either continuous or half-day; these two were classified as frequency A and frequency B respectively. Frequency A will be manipulating the household meter reading for one whole day and frequency B will only do this for half of the day, from 6AM to 6PM. This was done to add character to the pilferer since in real time pilferers usually steal electricity at their peak usage time.

With each of the 55 households simulated to be a pilferer, and with two varying frequencies of theft, the resulting data consists of $55 \times 2 = 110$ week-long simulations for each of the seven dataset categories. The resulting data represents the malicious data.

4.2.5 Features and Labelling

Dataset features determine the properties that describe the data points of a dataset. In the context of electricity theft detection, these features are usually customer time-series data derived from active energy consumption curves [19]. The following subsections will discuss the features for SVM, ANN, and Anomaly Coefficient Calculation. The difference between the features for SVM and ANN, and the features of Anomaly Coefficient Calculation will also be discussed.

4.2.5.1 Features for SVM and ANN

For theft detection algorithms SVM and ANN, the chosen features are the meter readings of each household and the % loss error of each check meter. Check meter readings are essentially the total power consumption of all downstream households, together with technical losses from the transmission lines. Table 4.2 shows which households are under each check meter, as derived from Figure 4.2. These check meters are usually out of reach from the public and thus, are rarely manipulated.

Check Meter	Households
1	1, 2, 3, 4, 5, 6
2	7, 8, 9, 10, 11, 12
3	13, 14, 15, 16, 17
4	18, 19, 20, 21, 22, 23
5	24, 26, 27, 28, 32, 38, 44
6	25, 29, 30, 31, 33, 34, 35, 36, 37
7	39, 40, 41, 42, 45, 46, 48, 49, 51
8	43, 47, 50, 52, 53, 55

Table 4.2: Check Meters and Respective Downstream Households

In a benign dataset, the check meter readings are almost exactly the same as the sum of the individual household meter readings; the small difference accounts for the technical losses from the transmission lines. On the other hand, in a malicious dataset where a pilferer manipulated their meter readings, the difference

between the check meter readings and the sum of the household meter readings can be significantly larger. The % loss error for a check meter *i* was calculated using

$$\%error_i = \frac{\left|\sum_{n=1}^{k} M_n - CM_i\right|}{CM_i} \tag{4.1}$$

where CM_i is the reading of check meter *i* and $\sum_{n=1}^{k} M_n$ is the sum of the household meter readings under check meter *i*. Thus, the features of the datasets for ANN and SVM are: the % loss error of the 8 check meters, and the meter readings of the 55 households; this totals to 63 features.

Every data point in each dataset category are the daily average values of the calculated % loss error and the week-long meter readings from Section 4.2.3 and Section 4.2.4. At the end of each data point, a label column was added with a value of:

- 0, if the data point is benign or there is no theft present
- 1, if the data point is malicious or theft is present

4.2.5.2 Features for Anomaly Coefficient Calculation Method

To satisfy the prerequisites of the Anomaly Coefficient Calculation method, the features that were selected differ from the ANN and SVM features. For this method, it was assumed that there is only a single check meter in the network and all households are under it. The altered topology is shown in Figure 4.6. Thus, the features for this algorithm are: the check meter readings of Check Meter 1 (main check meter), and the meter readings of the 55 households; this totals to 56 features.

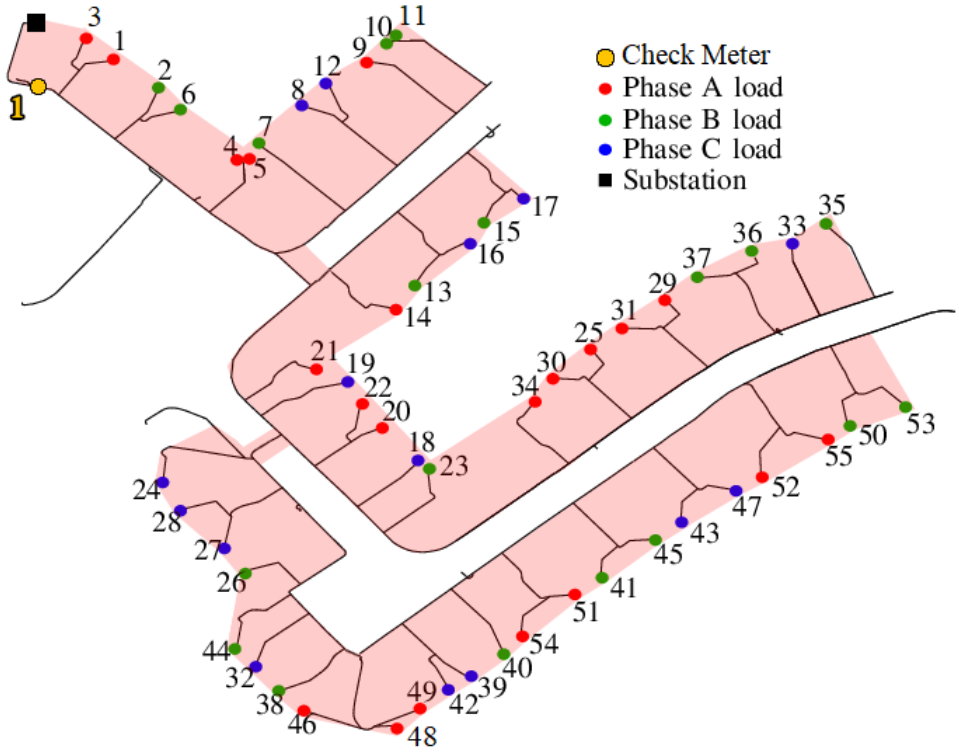


Image from [30]

Figure 4.6: Altered System Topology for Anomaly Coefficient Calculation Method

4.2.6 Dataset Summary

A summary of how each dataset is created is summarized below in Figure 4.7. For SVM and ANN, the features are the daily average meter readings and the % loss error of each check meter. There are a total of 63 features for SVM and ANN with one label. Since there are 55 households, 7 days per simulation, 2 theft frequencies, and 2 types of label, benign and malicious, there is a total of $55 \times 7 \times 2 \times 2 = 1540$ data points per dataset.

For Anomaly Coefficient Calculation, the features are presented as the total substation reading and the half-hourly meter readings. There are a total of 56 features and one label. Given that there are 55 households, with 48 half-hourly readings, 7 days per simulation, 2 theft frequencies, and 2 types of label, there is a total of $55 \times 48 \times 7 \times 2 \times 2 = 73920$ data points per dataset.

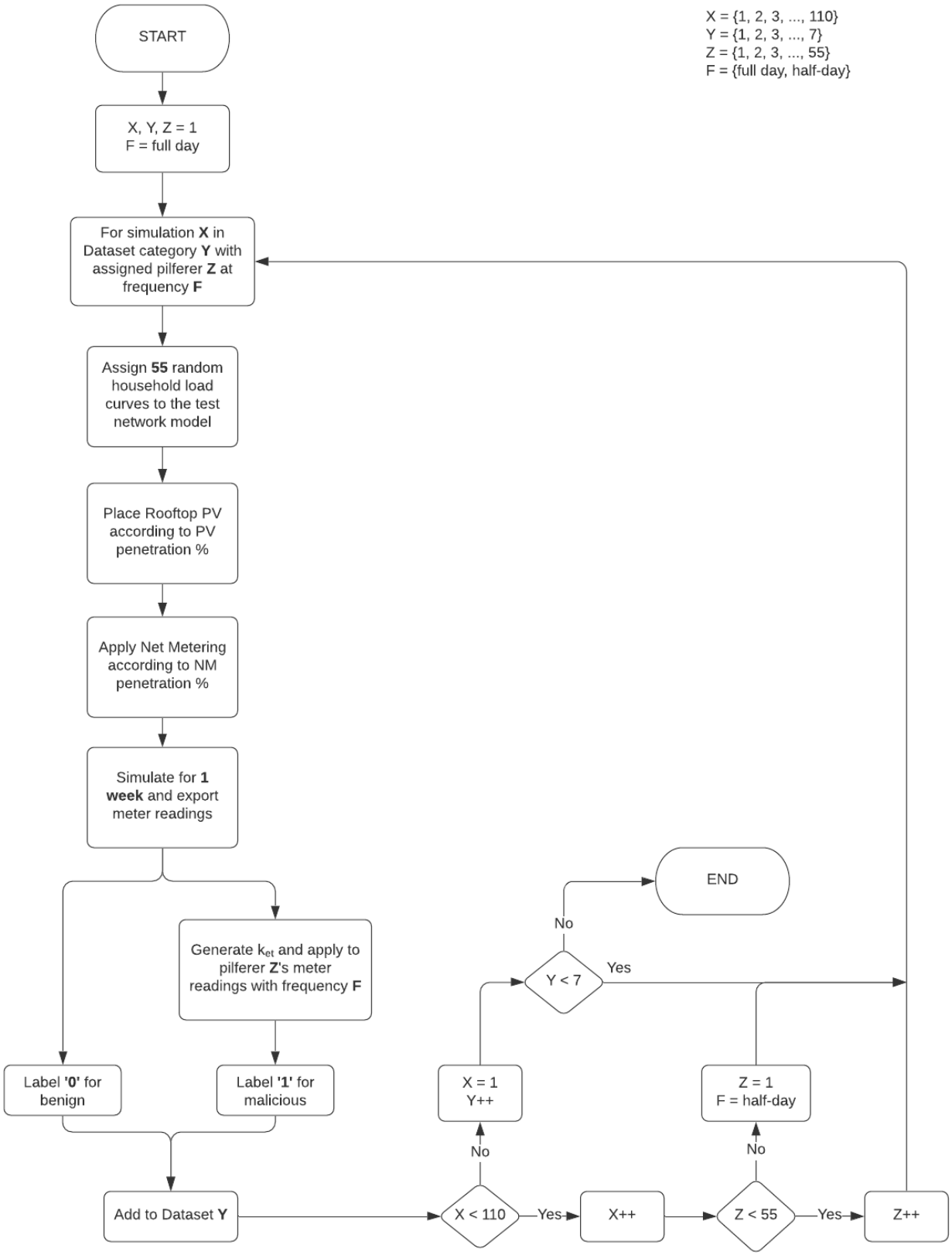


Figure 4.7: Dataset Creation Procedure

4.3 Implementation of Theft Detection Algorithms

The seven dataset categories created from Section 4.2 were used and evaluated on each of the three algorithms independently. The SVM classifier was developed using the Scikit-learn library, while the ANN classifier was developed using the Keras library. Both classifiers were written in Python using Jupyter Notebook and were executed using Google Colab Pro virtual machines for fast training and testing times. The third algorithm, Coincident Meter Measurements and Anomaly Coefficient Calculation, was developed using MATLAB.

4.3.1 SVM

An SVM classifier works by solving for a linear hyperplane that separates the classes with maximal margin. Figure 4.8 shows the procedure for SVM from [18].

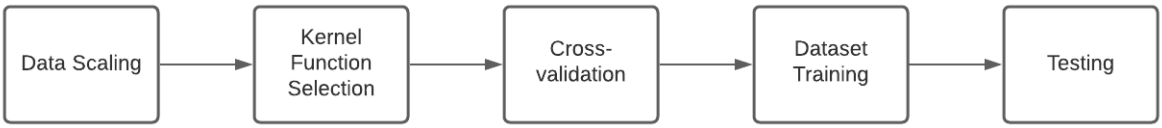


Figure 4.8: General Procedure for SVM

Initially, the datasets were standardized to a normalized scale such that the data have zero mean and unit variance. This was done to avoid large feature values dominating lower feature values [18]. The datasets were individually standardized using the following formula, where X' is the standardized value:

$$X' = \frac{X - \mu}{\sigma} \tag{4.2}$$

Each dataset was split 80:20 into training and testing sets. After the input data was standardized and split, a kernel function was selected based on the data structure. For this study, radial-basis function (RBF) kernel was chosen. According to [18], the RBF kernel is capable of handling cases in which the relationship between class labels and attributes are nonlinear.

To find the best parameters of the kernel function (C, γ), a grid-search using *n*-fold cross-validation was performed on the training set of each dataset category.

The *k*-fold cross-validation procedure involves dividing the training set into *n* subsets of equal size. Every iteration, the $n-1$ subsets were used to train the classifier and the remaining subset was used for validation. Figure 4.9 shows an example of a 5-fold cross-validation procedure.

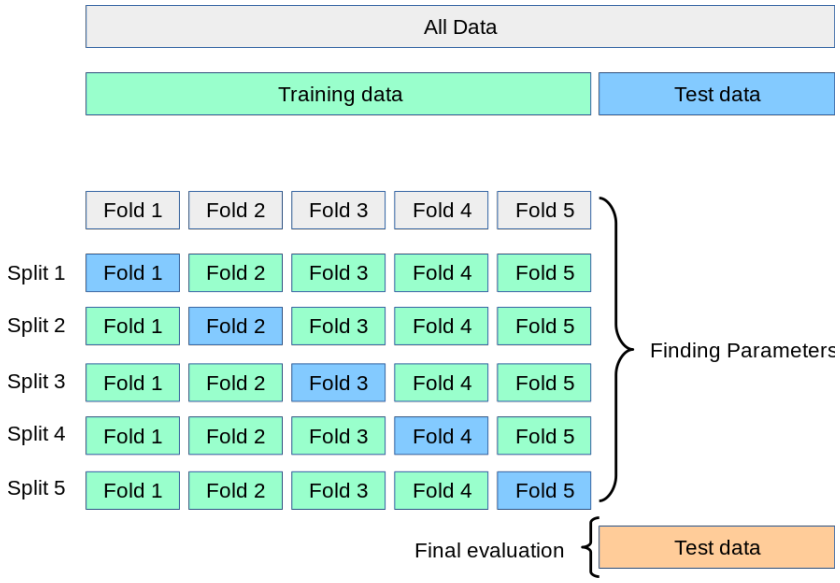


Image from [31]

Figure 4.9: 5-fold Cross-Validation

The grid-search method iterated through multiple combinations of (C, γ) pairs until the pair with the best cross-validation accuracy is achieved. The parameters were varied as follows:

$$C = [2^{-15}, 2^{-14}, 2^{-13}, ..., 2^{2}, 2^{3}]$$

$$\gamma = [2^{-15}, 2^{-14}, 2^{-13}, ..., 2^{2}, 2^{3}]$$

For this study, a 10-fold cross-validation was used in order to not overfit the training data [17]. Scikit-learn's built-in function GridSearchCV was used for this. After the best (C, γ) pair was obtained, the training set was trained again using the (C, γ) pair to obtain the final classifier. The performance of the classifier was evaluated using the testing set.

The entire training and testing procedure was repeated for each of the seven dataset categories in 4.1.

4.3.2 ANN

A deep feed forward neural network inspired by [26] was modeled using Keras, a deep-learning API. The neural network consisted of an input layer, multiple hidden layers, and an output layer. The input layer has 63 neurons, representing the features of the dataset: 55 household meter readings and % loss error of the 8 check meters. Let *L* be the number of hidden layers and *N* be the number of neurons in each of the hidden layers. The output layer has a single neuron that outputs a value of either '1', which means that theft is present, or '0', which means there is no theft present.

From Figure 4.10, it can be seen that for every neuron, there is a bias and a corresponding weight for every input. Let m be the number of inputs to a neuron, w_i as the weight of the *ith* input, and b as the bias value. The output of a single neuron is

$$y_{in} = b + \sum_{i=1}^{m} w_i x_i$$

An activation function maps y_{in} to a range of values to produce the neuron's final output. This is calculated for all neurons of each layer, from the input layer to the output layer.

Let W^l be the matrix that defines the weight of the neurons in layer *l*. Likewise, let b^l be the vector that defines the bias of neurons in layer *l*. Training the detector means using an optimization algorithm such as stochastic gradient descent to solve for the model parameters W^l and b^l by minimizing the cross-entropy loss function

$$\min_C C = -\left(y_d \log\left(p_d\right)\right) - \left(1 - y_d\right) \log\left(1 - p_d\right) \tag{4.3}$$

where y_d is the label of data point d and p_d is the prediction of the model for the same data point d. The model described in [26], which consists of a feed forward stage and a backward propagation stage every iteration, was used to solve for the optimal parameters W^l and b^l for every layer.

In designing neural networks, hyperparameters such as the optimizer, learning rate, activation function, and number of hidden layers and neurons, must be carefully selected such that the optimum performance is achieved [22]. In designing the neural network of this paper, grid-search using k-cross validation was performed to iteratively find the best combination of hyperparameters.

The subsequent sections discuss each of the hyperparameters used in training the neural network.

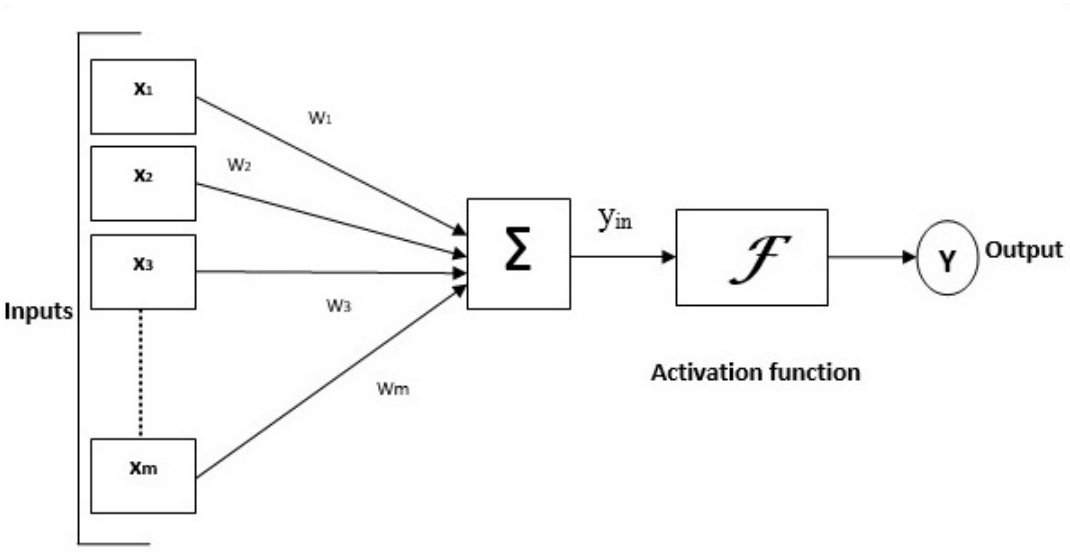


Image from [32]

Figure 4.10: Model of a single artificial neuron

4.3.2.1 Optimizer and Learning Rate

The optimization algorithm, $\mathcal{O}$, determines how the weights and biases of neurons in each layer are updated during the backpropagation stage. In choosing the best optimizer for the neural network, the computing time and its use in previous electricity theft detection research were considered. As such, the best optimizer was chosen between the Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (ADAM).

SGD is one of the most dominantly used optimization algorithms in deep-learning neural networks [33]. It is the preferred optimization algorithm for large datasets in which the computing time is the limiting factor rather than dataset size [34]. ADAM, another variant of gradient descent, is also typically used in large datasets but is more computationally efficient and requires little memory [33].

Both of these gradient-based optimizers have a parameter η called *learning rate*, which controls the step size or how fast the weights and biases are updated by the optimizer. Setting this too low will lead to slow training times, while setting it too high may cause divergence errors [22]. The learning rate is set to an initial value and gradually decreases during training. In practice, the initial learning rate is manually adjusted and depends highly on the problem, hence the need to find the optimal value through hyperparameter optimization [35].

For hyperparameter optimization, the following values were used:

$$\mathcal{O} = \{SGD, ADAM\}$$

$$\eta = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$$

4.3.2.2 Activation Function

In Figure 4.10, an activation function, $\mathcal{A}$, maps the output of a neuron to a range of values. In most neural networks, hidden layers use the same activation function while the activation function of the output layer highly depends on the problem. For binary classification problems, the sigmoid function is used because it maps the output to either 0 or 1.

For the hidden layers of the paper's neural network, the activation function was chosen between the sigmoid function, the hyperbolic tan function, and the Rectified Linear Unit (ReLU) function. These are the most commonly used activation functions in deep-learning models [36]. Figure 4.11 shows the plot of the three different activation functions.

The sigmoid function, given by

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4.4}$$

is a zero-centered function and maps the output between 0 and 1. It is typically used in output layers of binary classification models but can also be used in hidden layers as well.

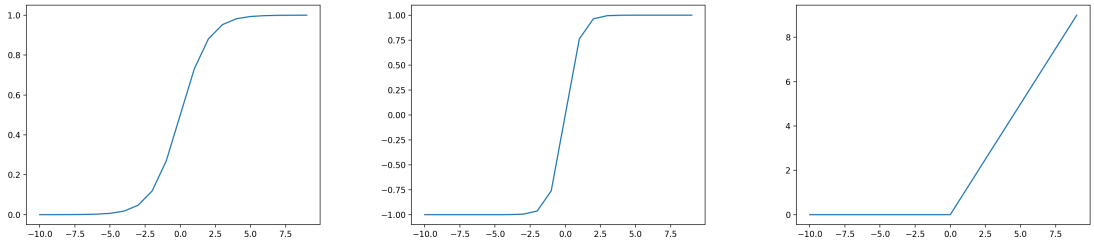
The hyperbolic tan function, given by

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.5}$$

is similar to the sigmoid function but has a smoother center and maps the output between -1 and 1. It is reported that the hyperbolic tan function gives better training performance for multi-layer neural networks [36].

The third activation function, ReLU, was developed in 2010 and is regarded as the most widely used activation function for hidden layers because it addressed the shortcomings of both the sigmoid and hyperbolic tan functions [36]. It is given by

$$f(x) = max(0, x) = \begin{cases} x_i, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{4.6}$$



(a) Sigmoid (b) Hyperbolic Tan (c) ReLU

Figure 4.11: Activation Functions

For hyperparameter optimization, the following values were used:

$$\mathcal{A}_{hidden} = \{sigmoid, tanh, relu\}$$

4.3.2.3 Epoch and Batch Size

The number of times the network trains on the training set is called epoch or $\mathcal{E}$. Every epoch, a small subset of the training set is used to update the weights and biases of the network. The size of this subset is called batch size or $\mathcal{B}$. Epoch and batch size are optimized together since in theory, they mostly affect training time and not necessarily performance. However, changes in epoch and batch size may affect other hyperparameters, and as such, must be optimized together with the other hyperparameters as well [22].

For hyperparameter optimization, the following values were used:

$$\mathcal{E} = \{20, 50, 100\}$$

$$\mathcal{B} = \{32, 64, 128\}$$

4.3.2.4 Hidden Layers and Number of Neurons

The number of neurons, or n_h, in each hidden layer, and the number of hidden layers, are directly correlated to the capacity and complexity of the neural network. For linear problems, a single hidden layer is enough, but for more complicated, non-linear problems, a network with multiple hidden layers perform better. According to [21], using the same number of neurons for all hidden layers produced better results than using different ones.

For this neural network, the number of hidden layers was set to 3. The number of neurons of each hidden layer was set to values approaching the maximum number of neurons in the input layers, as in [37].

For hyperparameter optimization, the following values were used:

$$n_h = \{20, 40, 60\}$$

4.3.2.5 *K*-fold Cross-Validation

The datasets were standardized to a normalized scale such that the data have zero mean and unit variance. Each dataset category in Table 4.1 was split 80:20 into training and testing sets.

Similar to Section 4.3.1, *k*-fold cross-validation was used to find the best performing combination of the hyperparameters discussed. Accuracy was used as the scoring criterion.

Figure 4.12 illustrates the neural network model to be optimized. The number written in a layer denotes the number of neurons in that layer. The following summarizes all the hyperparameters that were optimized using a 10-fold cross-validation procedure:

$$\begin{aligned}
\mathcal{O} &= \{SGD, ADAM\} \\
\eta &= \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\} \\
\mathcal{A}_{hidden} &= \{sigmoid, tanh, relu\} \\
\mathcal{E} &= \{20, 50, 100\} \\
\mathcal{B} &= \{32, 64, 128\} \\
n_h &= \{20, 40, 60\}
\end{aligned} \tag{4.7}$$

After obtaining the best performing combination of the hyperparameters, the training set was trained again using the chosen hyperparameters to obtain the final classifier. Finally, the performance of the classifier was evaluated using the testing set. The entire neural network was made using the Sequential model function from Keras.

The entire training and testing procedure was repeated for each of the dataset categories in 4.1.

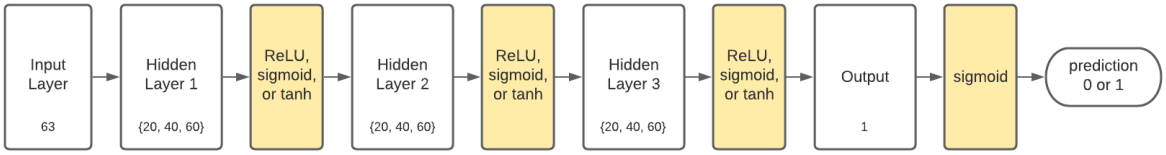


Figure 4.12: Theft Detection Neural Network Model

4.3.3 Anomaly Coefficient Calculation

Anomaly Coefficient Calculation describes the use of coincident meter measurements and writes them as sets of linear equations to detect pilferage [23]. By solving the linear equations, the algorithm derives a set of coefficients that is multiplied to each household meter and identifies the pilferer by analyzing the coefficients. The parameters are given by the coincident smart meter readings of the households and their respective check meter. The computed coefficients determine the factor by which the pilferer scaled its meter readings. The algorithm was originally tested on a test system where all customers are equipped with smart meters, and a check meter is connected upstream to measure the total consumption of the network.

A set of linear equations that are derived from Equation (4.8) can be formed from the check meter and household meter readings, which can be represented as a matrix equation $C = MK$. From the equation, having *N* meter readings would require *N* equations to solve for their respective anomaly coefficients *k*. The coefficients may be solved using matrix inversion if the matrix is square similar to Equation (4.9). Constrained least squares method is used if there are insufficient meter readings from a given system. Since a gathered time frame data could be less than the variables under study in a large test network, this method minimizes the errors in producing such coefficients for the system.

$$CM_t = M_{at}k_a + M_{bt}k_b + ... M_{nt}k_n \tag{4.8}$$

Where:

CM_t = check meter reading at time t

M_{xt} = meter reading of customer x at time t

k_xt = anomaly coefficient of customer x

$$\begin{bmatrix} k_a \\ k_b \\ k_c \\ \vdots \\ k_x \end{bmatrix} = \begin{bmatrix} M_{a1} & M_{b1} & M_{c1} & \dots & M_{x1} \\ M_{a2} & M_{b2} & M_{c2} & \dots & M_{x2} \\ M_{a3} & M_{b3} & M_{c3} & \dots & M_{x3} \\ \vdots & \vdots & \vdots & \ddots & \\ M_{at} & M_{bt} & M_{ct} & \dots & M_{xt} \end{bmatrix}^{-1} \times \begin{bmatrix} CM_1 \\ CM_2 \\ CM_3 \\ \vdots \\ CM_t \end{bmatrix} \tag{4.9}$$

Constrained least squares was used in this project since each data point has 48 time frames only, which is insufficient for a system with 55 households. Since *t*, number of time frames where theft occurs, is less than that of *x*, number of households inside the check meter, matrix inversion would be unable to solve for the corresponding coefficients of each household.

Constrained least square is the process of solving for *k* that may satisfy Equation 4.10. The upper and lower limits of the constraints used in this method is solved by using Equation 4.11 and inputting a maximum *m* value of 0.65, which is the maximum value of k_{et} derived from Section 4.2.4, and minimum *m* value of 0, in which there is no pilferage in the system. Using these constraints, the coefficients of each household can be solved while limiting it to an upper limit of 2.86 and a lower limit of 1. It is important to note that since there are not enough time frames to perform matrix inversion, the results may differ in cases wherein there are sufficient time frames.

$$MIN\|Ak - b\|^2 \tag{4.10}$$

Where:

A = matrix for the set of household readings

k = anomaly coefficient matrix

b = matrix of CM readings

$$k = \frac{1}{1+m} \tag{4.11}$$

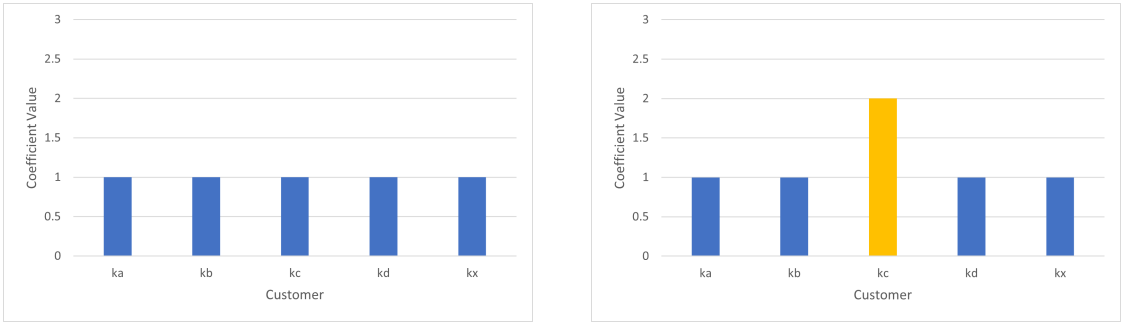
Where:

$+m$ means that the meter is recording **more** than expected

$-m$ means that the meter is recording **less** than expected

Given that there are sufficient time frames to work with, and there is no pilferer in the system, all the k in the equation will have a value close to one given that there is zero or minimal loss in the system. This means that the sum of all the smart meter readings is equal to the check meter reading. If a pilferer is present, their corresponding k will be greater than 1; this is illustrated in Equation 4.11, and shown in Figure 4.13.

In a real life setup, technical losses that are present in the system affect the error between the check meter and household meter readings. Because of this, the check meter readings may differ from the sum of the downstream households. Additionally, when the pilferage occurs at a very short duration, its effect on the system may be insignificant enough for the pilferer to be detected since the algorithms relies on the number of time frames with theft.



(a) k Without Pilferage

(b) k With Pilferage

Figure 4.13: Coefficient k Values

The algorithm's performance was tested using the datasets that were created in Section 4.2. Similar to [23], there is also one pilferer in each simulation of the created datasets. To highlight the effect of Net Metering and rooftop PVs on the the algorithm, the time frames wherein there is no pilferage present were removed. This is because inclusion of time frames with no pilferage will disturb the natural coefficient of each household as shown by [23]. For theft frequency B, these are the time frames outside 6AM to 6PM. Additionally, only the main check meter shown in Section 4.2.5.2 was used to replicate the original process done in [23]. Since there are 55 households and only 48 time frames, constrained least squares method was used to solve for the k values.

To identify the pilferer in the system, the calculated k value of each household were compared to a threshold value. The pilferer is identified if its k value is larger than the threshold value. Setting the threshold value is a crucial part of the algorithm, and significantly affects its ability to correctly detect and identify the pilferer.

Because of this, multiple thresholds were considered and tested. The different thresholds were computed based on the possible values of k_{et} in Section 4.2.4 and the Equation 4.11. If there are multiple k values that are larger than the threshold, the largest k and its corresponding household was predicted to be the pilferer.

The outputs of this algorithm are: whether or not there is pilferage in the system, and the household number of the pilferer.

4.4 Performance Metrics Calculation

For binary classification models, a two-by-two confusion matrix determines the four possible outcomes of the predicted value of the classifier. From the confusion matrix in 4.3, the four measures that can be obtained are: True Positive (TP) and True Negative (TN), which are the number of correctly predicted positive and negative cases, and False Positive (FP) and False Negative (FN), which refers to the number of wrongly predicted positive and negative cases [38][39].

Actual Value	**Predicted Value**	
	Positive	**Negative**
Positive	True Positive (TP)	False Negative (FP)
Negative	False Positive (FP)	True Negative (TN)

Table 4.3: Confusion Matrix

For evaluating the theft detection and theft identification algorithms, accuracy, precision, recall, and F1 scores were used. These metrics were the ones considered on whether Rooftop PV and Net Metering can significantly affect the performance of electricity theft detection and identification algorithms.

4.4.1 Accuracy

This gives the ratio of correctly predicted observations (TP + TN) to the total observations. While accuracy is a great way of determining outright the best model, the symmetry of the datasets is also to be considered. Accuracy works best for datasets that are symmetric where values of false positives (FP) and

false negatives (FN) are almost the same [40]. The formula for accuracy is given by

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.12}$$

4.4.2 Precision

This is the ratio of correctly predicted positive observations (TP) to the total predicted positive observations, may it be correct (TP) or incorrect (FP). High scores of precision relate to a low false positive rate. Precision can be seen as a measure of the quality of the classifier. The formula for precision is given by

$$Precision = \frac{TP}{TP + FP} \tag{4.13}$$

4.4.3 Recall

Recall, sometimes referred to as sensitivity, determines the ratio of correctly predicted positive observations (TP) to all actual positives (TP + FN). High recall means that an algorithm returns most of the relevant results, whether or not irrelevant ones are also returned. Unlike the precision metric earlier, this is more of a quantity measurement of the classifier. The formula for recall is given by

$$Recall = \frac{TP}{TP + FN} \tag{4.14}$$

4.4.4 F1

F1 is the weighted average of Precision and Recall. This gives us a more in-depth result than the Accuracy as it takes into account both false positive (FP) and false negative (FN). This is usually a better metric especially if false positive (FP) and false negative (FN) do not bear the same weight, especially on datasets that are not symmetric [40]. The formula for F1 score is given by

$$F1Score = \frac{2(DR \cdot PR)}{DR + PR} \tag{4.15}$$

Chapter 5

Results and Discussions

5.1 Theft Detection Algorithm Results

5.1.1 SVM Results

After performing 10-fold cross-validation on the training sets of each dataset category, the (C, γ) pairs with the highest accuracy scores were selected. The optimum parameter pairs of each dataset category are presented in Table 5.1.

Dataset	**C**	γ
D1 (0% PV)	8	0.015625
D2 (20% PV)	8	0.015625
D3 (70% PV)	8	0.0078125
D4 (20% PV, 20% NM)	8	0.0078125
D5 (20% PV, 70% NM)	8	0.0078125
D6 (70% PV, 20% NM)	8	0.0078125
D7 (70% PV, 70% NM)	8	0.0078125

Table 5.1: SVM Parameter Results

Using the optimized parameters, the classifiers were trained again on the training set and evaluated on the testing set of each dataset category. Since there is an element of randomness during the training stage, testing and training were repeated 6 times per dataset and the average of its performance metrics

were recorded. Table 5.2 presents the final performance metrics of each dataset using an SVM classifier. A single run of training and testing took approximately 12 minutes.

Dataset	Accuracy	Precision	Recall	F1
D1 (0% PV)	0.9708	0.9989	0.9426	0.9699
D2 (20% PV)	0.9654	0.9943	0.9361	0.9643
D3 (70% PV)	0.9275	0.9819	0.8712	0.9230
D4 (20% PV, 20% NM)	0.9156	0.9772	0.8517	0.9097
D5 (20% PV, 70% NM)	0.8312	0.9340	0.7132	0.8078
D6 (70% PV, 20% NM)	0.7549	0.8792	0.5909	0.7054
D7 (70% PV, 70% NM)	0.5985	0.6724	0.3842	0.4881

Table 5.2: SVM Results

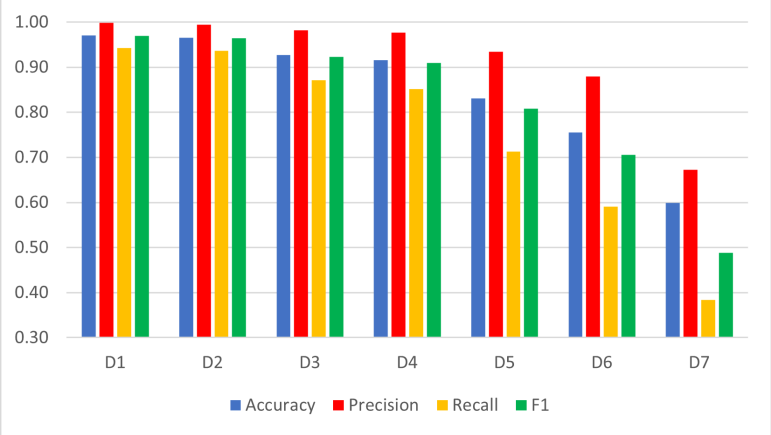


Figure 5.1: SVM Results - Dataset 1 to 7 Graph

The average accuracy across all datasets is 85.20%, which shows that the SVM classifier can successfully detect electricity theft given the meter readings of each household and the % loss error of each check meter. Dataset 1 has the highest metric scores with an accuracy of 97.08%, while Dataset 7 has the lowest with an accuracy of 59.85%. Figure 5.1 presents the overall trend across the datasets. The graph shows that as the number of households with rooftop PV and Net Metering increases, the metric scores worsen.

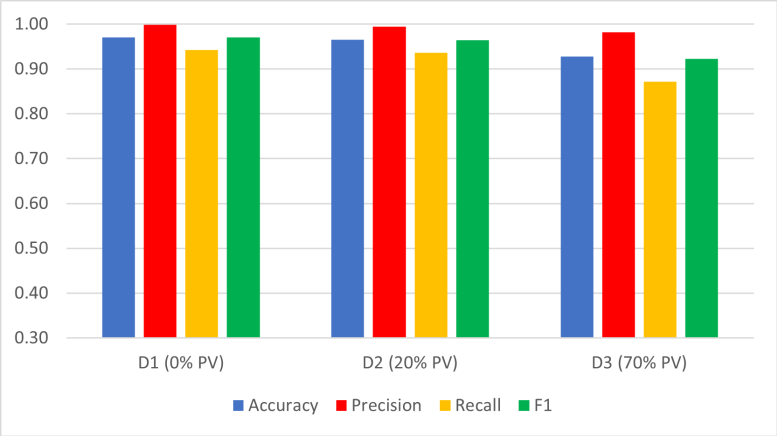
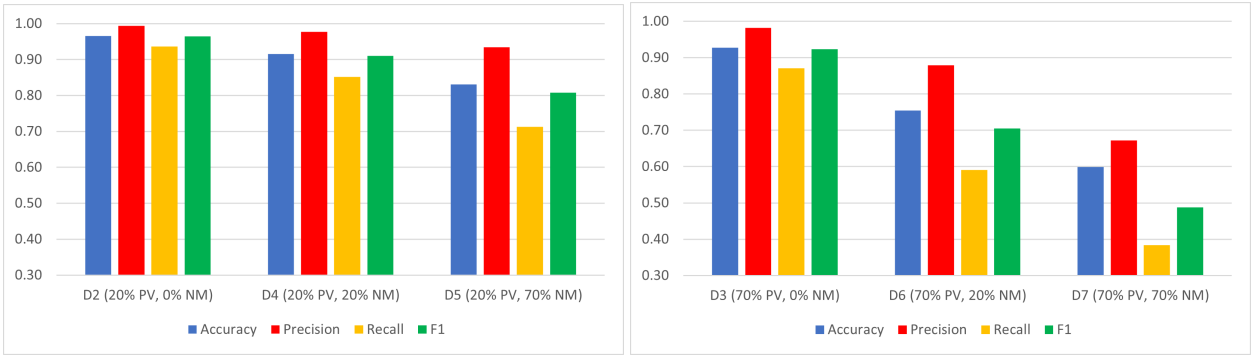


Figure 5.2: SVM Results - Varying PV Penetration Graph

Figure 5.2 compares Datasets 1, 2, and 3, which has varying PV penetration and no Net Metering. The metric scores of these datasets have very close values, with the accuracy scores having a low standard deviation of 1.93%. Despite having very close values, there is still a slightly noticeable downward trend in classifier performance as PV penetration is increased.



(a) 20% PV Penetration

(b) 70% PV Penetration

Figure 5.3: SVM Results - Constant PV Penetration with Varying NM Penetration Graph

Figures 5.3 presents the datasets in which the PV penetration is held constant at 20% and 70% respectively, while the NM penetration is varied. At a constant 20% PV penetration, there is very noticeable downward trend as Net Metering penetration is increased. The accuracy scores of Dataset 2, 4, and 5, have a standard deviation of 5.54%. This is slightly higher than when only the PV penetration is varied and there is no Net Metering present. When the PV penetration is increased to 70%, the downward trend becomes steeper as the Net Metering penetration is varied; the standard deviation of the accuracy scores increased to

13.44%. To illustrate, given a network with 55 households and at 20% PV and 20% Net Metering, the number of households with PV is $\lfloor 55 \times 0.2 \rfloor = 11$ and the number of households with both PV and Net Metering is $\lfloor 11 \times 0.2 \rfloor = 2$. At 70% PV and 20% Net Metering, the households with PV is now $\lfloor 55 \times 0.7 \rfloor = 38$ and the number of households with both PV and Net Metering is $\lfloor 38 \times 0.2 \rfloor = 11$. Because the number of households with PV increased drastically at 70% PV penetration, the effect of Net Metering is amplified, hence the worsened metrics scores.

Based on the simulation results, adding rooftop PV and Net Metering to a network has a conclusive effect on the performance of SVM electricity theft detectors that primarily use household meter readings and check meter % loss error as features. The difference between a worst-case scenario network having a majority of its households with both rooftop PV and Net Metering, and a network with no rooftop PV at all, is very significant. Between Datasets 1 and 7, accuracy drastically dropped by 37.23%. If having an accuracy of 80% is used as the standard in evaluating a classifier as effective, then using SVM on a network with up to 20% of its households installed with rooftop PV and up to 70% of those installed with Net Metering, will produce successful predictions. Further increasing the penetration of both PV and Net Metering worsens the capability of the classifier to detect electricity theft.

5.1.2 ANN Results

Presented in Table 5.3 are the optimized hyperparameters of the neural network for each of the datasets after performing 10-fold cross-validation. Almost all of the datasets picked SGD as the best optimizer except Dataset 6 and 7. For the hidden layer activation function, ReLU was picked for all datasets. All the datasets, except Dataset 1, took 100 epochs until the best accuracy score was achieved. Dataset 1 trained considerably faster; it only took 50 epochs for it to achieve its best accuracy score. Dataset 7 has the highest learning rate, η, at 0.9, which made sense since its batch size $\mathcal{B}$ is also the highest at 128. Higher batch sizes typically require faster learning rates to achieve the same accuracy score [22].

Dataset	$\mathcal{O}$	η	$\mathcal{A}_{hidden}$	$\mathcal{E}$	$\mathcal{B}$	n_h
1	SGD	0.1	ReLU	50	32	60
2	SGD	0.1	ReLU	100	64	40
3	SGD	0.1	ReLU	100	64	40
4	SGD	0.1	ReLU	100	64	40
5	SGD	0.1	ReLU	100	32	60
6	ADAM	0.3	ReLU	100	64	60
7	ADAM	0.9	ReLU	100	128	60

Table 5.3: ANN Optimized Hyperparameters

Dataset	**Accuracy**	**Precision**	**Recall**	**F1**
D1 (0% PV)	0.9800	0.9978	0.9621	0.9796
D2 (20% PV)	0.9735	0.9814	0.9654	0.9733
D3 (70% PV)	0.9226	0.9264	0.9199	0.9225
D4 (20% PV, 20% NM)	0.9188	0.9387	0.8961	0.9169
D5 (20% PV, 70% NM)	0.8696	0.8814	0.8571	0.8679
D6 (70% PV, 20% NM)	0.8009	0.8413	0.7435	0.7885
D7 (70% PV, 70% NM)	0.6374	0.6856	0.5238	0.5874

Table 5.4: ANN Results

After obtaining the optimum hyperparameters for each dataset, the classifiers were trained again on the training set and evaluated using the testing set. This was performed 6 times per dataset and the average of the performance metrics were recorded. Table 5.4 shows the measured performance metrics of each dataset. The neural network was modeled in Keras and training and testing time took approximately 8 minutes per run.

The results show that all datasets except Dataset 7 have accuracy scores above 80%. The average accuracy across all datasets using a neural network classifier is 87.18%, with a standard deviation of 11.13%. It is worth noting that the base case, Dataset 1, has almost perfect metric scores. Its precision is almost 100%, signifying that false positive outcomes are very rare. Additionally, its recall is less than the precision, which tells us that making a false negative is more common than making a false positive. This is satisfactory because it is less possible for the distribution utility company to accuse innocent customers.

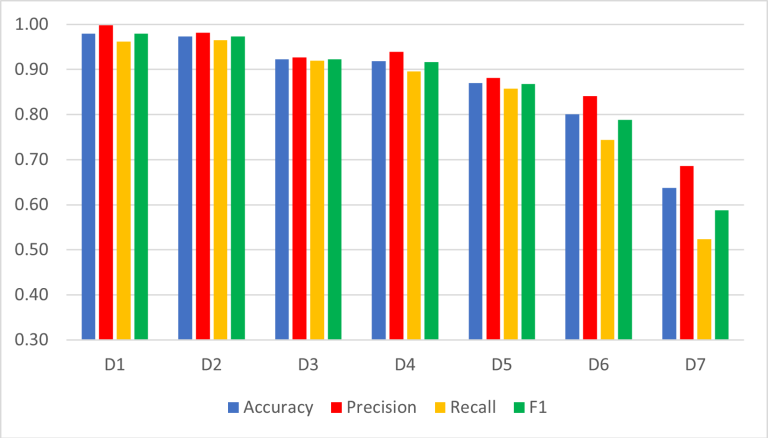


Figure 5.4: ANN Results - Dataset 1 to 7 Graph

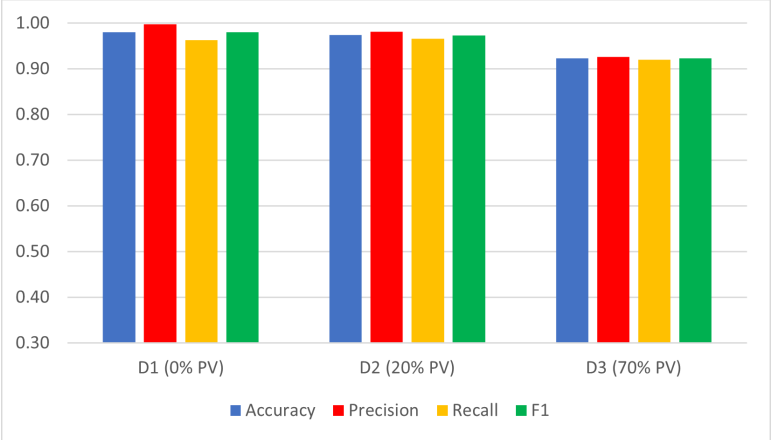
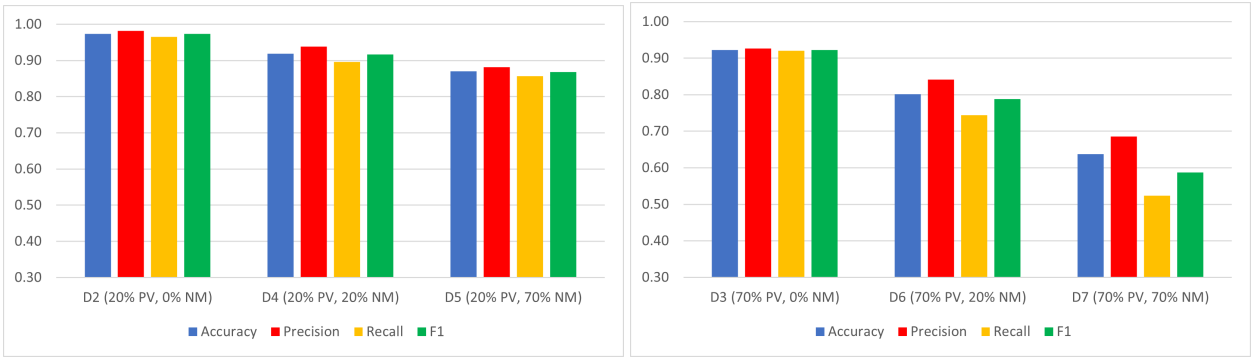


Figure 5.5: ANN Results - Varying PV Penetration Graph

The overall trend of the performance metrics across datasets is highlighted in Figure 5.4. Datasets 1, 2, 3 and 4, have high metric scores above 90%, while Dataset 7 has the worst scores. The recall of Dataset 7 is 52.38%, which is alarming because it signifies that it has a high number of false negative predictions. A pilferer would have the best chances of staying undetected in a network with a majority of its households installed with both PV and Net Metering.

Figure 5.5 shows that at varying PV penetration with no Net Metering, the difference in the metric scores is minimal. On the other hand, Figure 5.6 shows that if PV penetration is held constant and we vary the number of households with Net Metering, the metric scores decrease significantly. The effect of Net Metering is more apparent in Figure 5.6b since the base number of households with PV is higher (70% PV penetration). Therefore, Net Metering has a worse effect on the performance of a neural network classifier

than the presence of PV alone.



(a) 20% PV Penetration

(b) 70% PV Penetration

Figure 5.6: ANN Results - Constant PV Penetration with Varying NM Penetration Graph

Based on the results of the simulations, it is evident that a neural network classifier trained using a dataset with household meter readings and % loss error as its features, will perform exceptionally in detecting electricity theft, provided that the majority of the households do not have both PV and Net Metering installed. As the number households with Net Metering increases, performance of the classifier worsens especially at high PV penetration.

It is worth noting that the results of the SVM classifier is very similar to that of the ANN classifier. The overall trend across varying levels of PV and Net Metering penetration is the same on both types of classifiers: detection performance has a negative relationship with PV and Net Metering penetration. The similarity in the results can be attributed to the fact that both types of classifiers used the same dataset with the same features. As such, it is necessary to examine the characteristics of the dataset and its features to identify the cause of the decreasing trend.

5.1.3 Anomaly Coefficient Calculation Results

Anomaly Coefficient Calculation identifies the household where the theft occurs unlike SVM and ANN. Because of this, its confusion matrix is different from the confusion matrix of SVM and ANN. The way to determine a True Positive, True Negative, False Positive, or False Negative is given by:

- TP, if the algorithm correctly predicts that there is theft and identifies the right pilferer.
- TN, if the algorithm correctly predicts that there is no theft.

- FP, if the algorithm correctly predicts that there is theft, but identifies the wrong pilferer. Or, the algorithm wrongly predicts that there is theft but there is no theft.
- FN, if the algorithm wrongly predicts that there is no theft but there is theft.

To make sure that the best threshold was selected for each dataset, its value was varied and the resulting performance metrics were compared with each other. The threshold 1.25 gave the highest accuracy, precision, recall and F1 score.

Since the simulations with theft frequency B have less time frames where theft occurs, it was suspected that the algorithm may perform poorly when it is tested on this part of the dataset. With regards to this, the seven datasets were split into two subsets to further check the accuracy of the algorithm when different frequencies are tested. Three different testing data were used: the original dataset with frequencies A and B, the subset with frequency A, and the subset with frequency B.

With a threshold of 1.25, the results were recorded and tabulated. The tables are separated with respect to the included data per test. Tables 5.5, 5.6, and 5.7 show the metric scores of each case. Shown below in Figure 5.7, 5.8, and 5.9 are the bar graph results of the tables.

Dataset	Accuracy	Precision	Recall	F1
D1 (0% PV)	0.8429	0.8159	0.8208	0.8183
D2 (20% PV)	0.8273	0.7703	0.8262	0.7973
D3 (70% PV)	0.8318	0.7737	0.8571	0.8133
D4 (20% PV, 20% NM)	0.7935	0.7311	0.7756	0.7527
D5 (20% PV, 70% NM)	0.8513	0.8368	0.8258	0.8312
D6 (70% PV, 20% NM)	0.8266	0.7527	0.8729	0.8083
D7 (70% PV, 70% NM)	0.8422	0.7945	0.8618	0.8268

Table 5.5: Anomaly Coefficient Calculation Results (Both Frequencies)

Dataset	Accuracy	Precision	Recall	F1
D1A (0% PV)	0.9506	0.9777	0.9213	0.9486
D2A (20% PV)	0.9351	0.9605	0.9043	0.9315
D3A (70% PV)	0.9468	0.9616	0.9286	0.9448
D4A (20% PV, 20% NM)	0.9273	0.9793	0.8711	0.9220
D5A (20% PV, 70% NM)	0.9338	0.9883	0.8779	0.9298
D6A (70% PV, 20% NM)	0.9416	0.9538	0.9261	0.9398
D7A (70% PV, 70% NM)	0.9364	0.9563	0.9138	0.9346

Table 5.6: Anomaly Coefficient Calculation Results (Frequency A)

Dataset	Accuracy	Precision	Recall	F1
D1B (0% PV)	0.7351	0.6278	0.6855	0.6554
D2B (20% PV)	0.7195	0.5631	0.7121	0.6289
D3B (70% PV)	0.7169	0.5852	0.7607	0.6615
D4B (20% PV, 20% NM)	0.6597	0.4722	0.6270	0.5387
D5B (20% PV, 70% NM)	0.7688	0.6807	0.7584	0.7175
D6B (70% PV, 20% NM)	0.7117	0.5579	0.7970	0.6563
D7B (70% PV, 70% NM)	0.7481	0.6319	0.7931	0.7034

Table 5.7: Anomaly Coefficient Calculation Results (Frequency B)

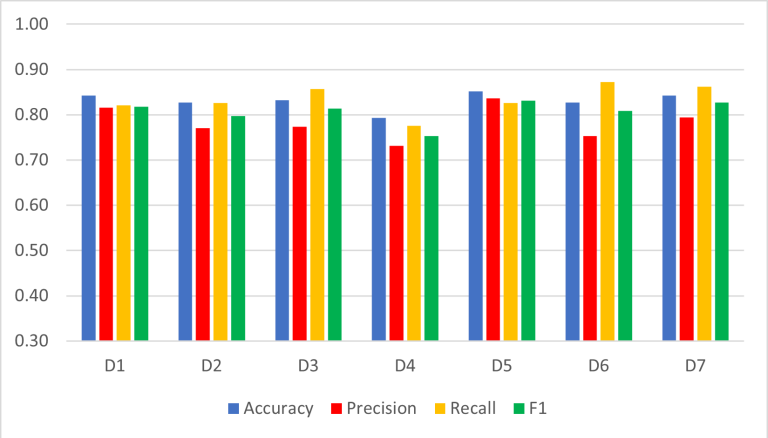


Figure 5.7: Anomaly Coefficient Calculation Graph (Both Frequencies)

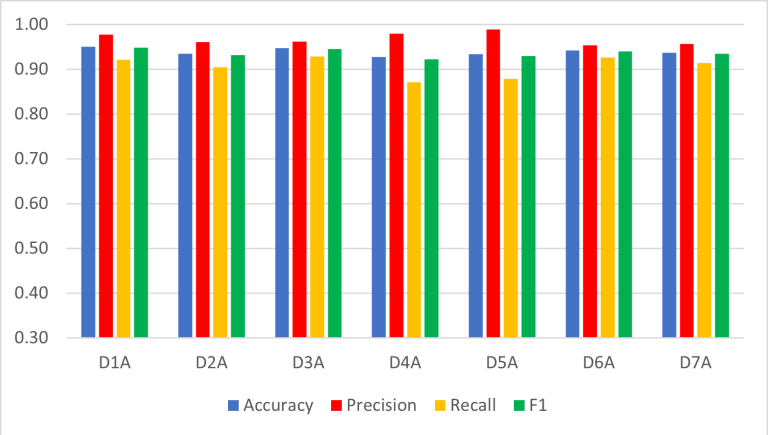


Figure 5.8: Anomaly Coefficient Calculation Graph (Frequency A)

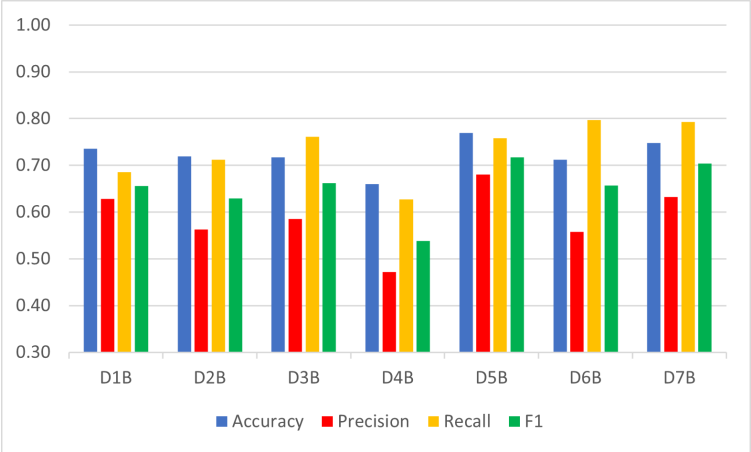


Figure 5.9: Anomaly Coefficient Calculation Graph (Frequency B)

It can be observed that the results from Figure 5.8 scored the highest, with Figure 5.7 as the second highest, and Figure 5.9 as the lowest. From the results, it can be inferred that the metric scores of this method is significantly affected by the ratio of the number of time frames with theft to the number of households.

Since the datasets have the same number of benign and malicious points, making it symmetric, and False Negative (FN) and False Positive (FP) are valued the same. Therefore, the accuracy would be the best metric to check. Looking at the accuracy, the performance metric data as shown in Table 5.5 and 5.6 both show a successful metric as a classifier, both being above 80%. The algorithm performed well with an average accuracy of 83.08% for the dataset that contains both frequencies and an average accuracy of 93.88% for the dataset that contains only frequency A. However, it can be seen in Table 5.7 that the case with only Frequency B did not have the same success as seen in the other cases. Having only an average of 72.28%, it fails to achieve an accuracy of at least 80%.

As seen in the Figures 5.7, 5.8, and 5.9, unlike in SVM and ANN, there is no decreasing trend between the performance metrics of the seven different datasets in each case. This suggests that varying PV and Net Metering penetration do not have a clear effect on Anomaly Coefficient Calculation. The reason behind this is attributed to the fact that the introduction of PV and Net Metering arithmetically adds and subtracts values to both sides of the linear equations (Equation 4.9). The PV generation of a household will be 'deducted' from its meter reading, and the same value will also be subtracted from check meter reading. For houses with Net Metering, the same reason applies, the only difference is that the meter reading of the household may become negative. Therefore, the equation that is used to solve for k remains unchanged.

The slight differences in the anomaly coefficients k are attributed to the randomness of the datasets during the simulations, since each of the seven datasets used different load curves and PV generation curves. This also explains the smalls differences in accuracy, precision, recall and F1 scores across the different datasets.

5.2 Comparison of Algorithm Results

Theft detection methods using SVM and ANN detect whether theft is present in the system. On the other hand, Anomaly Coefficient Calculation not only detects theft, but also pinpoints which household is the pilferer. Because of this, the metric scores of SVM and ANN cannot be compared directly to that of the Anomaly Coefficient Calculation.

To identify the cause of the decreasing trend in classifier performance of both ANN and SVM when PV and Net Metering penetration is increased, the characteristics of the datasets were examined. When theft is present, the most obvious fact is that the % loss error of the check meter that the pilferer belongs to increased significantly.

In order to examine how the percent error values change as PV and NM penetration are varied, the distribution is plotted using histograms. This also illustrates the possible values of the % loss error of every dataset.

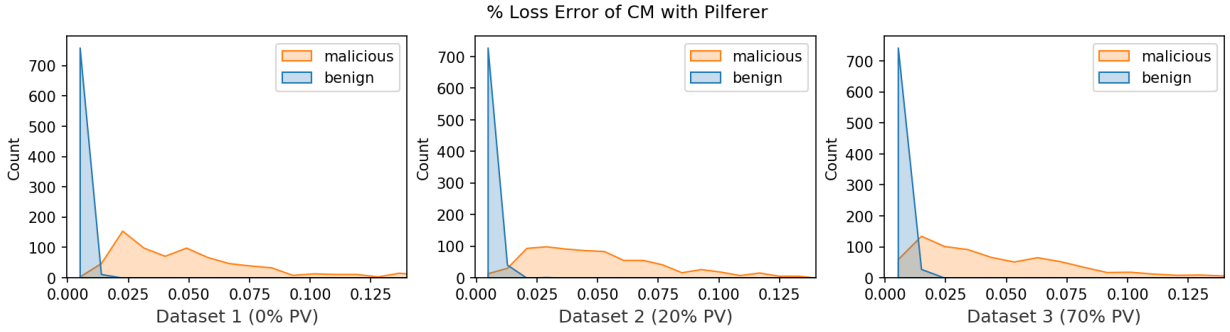


Figure 5.10: Histogram of % Loss Error for Datasets 1, 2, and 3

Figure 5.10 shows the histogram plots of the % Loss Error of the check meter in which the pilferer is present using Datasets 1, 2, and 3. Note that for Datasets 1, 2, and 3, only the PV penetration is varied at 0%, 20%, and 70% penetration respectively. The y-axis represents the number of occurrences while the x-axis represents the % loss error value. For the three datasets, the benign data points lie mostly on the postive values most near to 0; the slight deviation from 0 is attributed to technical losses caused by the transmission lines. When theft is present, the % loss error increases and moves further to the right. This is because the pilferer decreases their smart meter readings significantly which causes large deviation from the check meter readings, hence the increase in % loss error. However, as the PV penetration is increased, the distribution of the % loss error of the malicious data moves closer to that of the benign data. The distance between the mode of the malicious data and the mode of the benign data decreases which creates an overlap in their distribution. This made it harder for the SVM and ANN classifiers to separate the two classes. Notice that the amount of overlap in Dataset 1 and 2 are almost the same; this is supported by the similar metric scores of Datasets 1 and 2 in both ANN and SVM. In SVM, the accuracy scores of Dataset 1 and 2 are 97% and 96.5%, respectively. While in ANN, the accuracy scores are 98% and 97.3%, respectively. The small amount of households installed with PV in Dataset 2 has little to no effect in the performance of the classifiers.

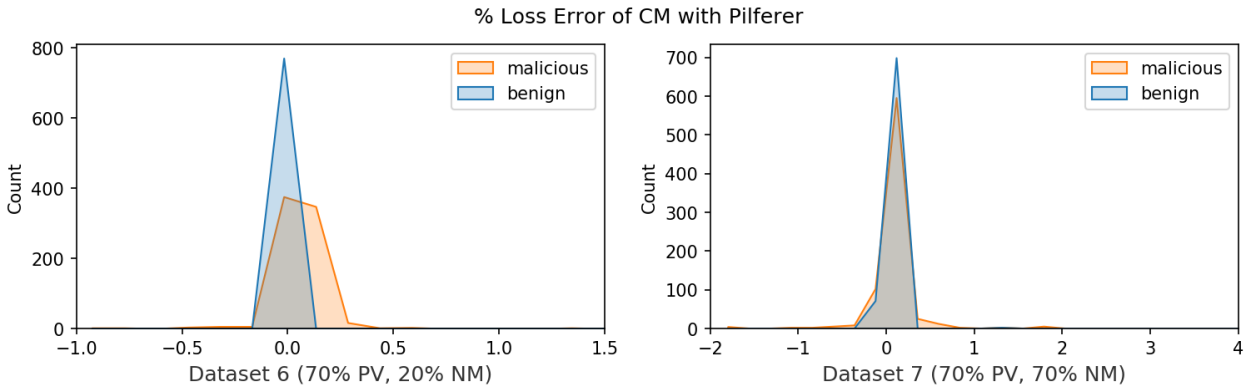


Figure 5.11: Histogram of % Loss Error for Datasets 6 and 7

To characterize the effect of Net Metering, the histogram plots of Datasets 6 and 7 are compared. Figure 5.11 shows the histogram plot of Datasets 6 and 7, in which PV penetration is constant at 70% while Net Metering penetration is at 20% and 70%, respectively. To reiterate, the formula for % loss error of a check meter is given by

$$\%error_i = \frac{\left|\sum_{n=1}^{k} M_n - CM_i\right|}{CM_i}$$

where CM_i is the reading of check meter *i* and $\sum_{n=1}^{k} M_n$ is the sum of the household meter readings under check meter *i*. The distribution plots shows that there could be negative values of % loss error for benign data points. This only happens when the net reading of the check meter is negative, which means that the grid receives power from the loads instead of delivering it: the total generated energy exceeds the total consumption of the nearby households. This usually occurs when most of the households under the check meter uses Net Metering.

In Dataset 6, most of distribution curve of the malicious data points overlap that of the benign data points, while in Dataset 7, the malicious data points overlaps the benign data points almost completely. In these two datasets, the classifier's performance is drastically reduced because it cannot separate the two classes anymore; the accuracy scores of Dataset 7 for SVM and ANN are 59.85% and 63.74% respectively, the lowest out of all datasets. In a system that has a high PV and NM penetration, the pilferer's activity is harder to detect because the possible range of values for the % loss error is widened.

High positive values of % loss error in both the benign and malicious data points happen when the net reading of the check meter is a small positive number, which means that the grid is delivering a very small amount of energy to the households. In this scenario, the total consumption of the nearby households is only slightly greater than the total generated energy of the PVs.

On the other hand, when a pilferer is present, the % loss error becomes a large negative value. This is the scenario in which the pilferer is a household with an already negative meter reading (production is much larger than consumption). When theft is applied, k_{et} from Section 4.2.4 is multiplied to an already negative value. This action will translate to a customer increasing their actual rooftop PV generation. This could happen in real life because the main incentive of Net Metering is that excess energy exported to the grid is paid for by the distribution utility company.

Therefore, it is evident that the % loss error of check meters is not the best feature to use in developing electricity theft classifiers for networks that has a high number of households with PV and Net Metering installed. High penetration values of PV and Net Metering alters the distribution curve of the % loss error and consequently, the classifier's ability to generalize is worsened hence the decreasing trend of performance metrics. Based on the results, Net Metering penetration affects the performance of the classifiers more than PV penetration alone. This is because of the increased possibility of negative check meter readings which affects the % loss error calculation.

For networks with low PV penetration, the use of % loss error as a feature is more than capable in detecting electricity theft. The main advantage of using this kind of feature is its intuitiveness and simplicity; it is easily calculated using check meter and household meter readings which are readily accessible measurements.

Chapter 6

Conclusion and Recommendations

This study explored the effect of rooftop PVs and Net Metering on theft detection algorithms. The results showed that the presence of PV and Net Metering in the system significantly affected the performance of SVM and ANN trained on datasets with % loss error of check meters and household smart meter readings as its features. Due to the nature of how % loss error is calculated, the presence of PV and Net Metering alters and shifts the distribution of the % loss error values which makes it harder for the SVM and ANN classifiers to distinguish between benign and malicious data.

With the use of % loss error as the main feature for datasets trained on SVM and ANN, increasing PV and Net Metering penetration caused a decreasing trend in the metric scores of these algorithms. At low levels of PV penetration, the algorithms performed outstandingly, with accuracy scores as high as 98% for ANN and 97% for SVM. However, systems with very high Net Metering penetration performed the worst with accuracy scores as low as 64% for ANN and 60% for SVM. Despite its intuitiveness and accessibility, using % loss error of check meters as one of the dataset features is not the best choice especially in systems with high PV and Net Metering penetration.

For the Anomaly Coefficient Calculation method, results show that the presence of PV and Net Metering does not affect the performance of the algorithm. It was inferred that the small differences in accuracy across the datasets are caused by the randomness of the input load and generation curves. Having lesser time frames or meter readings result in a drastic decrease in accuracy. It is best to use this algorithm when the given data has sufficient data points.

For future works regarding SVM and ANN, it is recommended to identify better features that can generalize data regardless if PV or Net Metering is present. The performance of these algorithms can also be

explored further by introducing systems with different forms of theft. Different theft detection algorithms can be explored that are capable of identifying the pilferer even in cases where the theft frequency is irregular.

Bibliography

[1] DOE, *Philippine power situation report 2019*, https://www.doe.gov.ph/sites/default/files/pdf/electric_power/2019-power-situation-report.pdf Accessed: 1-5-2021, 2019.

[2] ——, *Philippine power situation report 2014*, https://www.doe.gov.ph/sites/default/files/pdf/electric_power/power_situationer/2014_power_situationer.pdf Accessed: 1-5-2021, 2014.

[3] D. G. fur Internationale Zusammenarbeit, *More sun in the philippines - facts and figures on solar energy in the philippines project development programme (pdp) southeast-asia*, https://www.doe.gov.ph/sites/default/files/pdf/netmeter/policy-brief-its-more-sun-in-the-philippines-V3.pdf Accessed: 1-5-2021, 2013.

[4] *Republic act no. 9513 - an act promoting the development, utilization and commercialization of renewable energy resources and for other purposes*, https://www.officialgazette.gov.ph/2008/12/16/republic-act-no-9513 Accessed: 1-5-2021, Republic of the Philippines, Jul. 2008.

[5] A. Poullikkas, G. Kourtis, and I. Hadjipaschalis, "A review of net metering mechanism for electricity renewable energy sources.," *International Journal of Energy & Environment*, vol. 4, no. 6, 2013.

[6] Encorsolar, "What is net metering? how does it benefit me?.," *https://encorsolar.com/what-is-net-metering-how-does-it-benefit-me/*, 2019.

[7] P. Glauner, A. Boechat, L. Dolberg, R. State, F. Bettinger, Y. Rangoni, and D. Duarte, "Large-scale detection of non-technical losses in imbalanced data sets," Sep. 2016. DOI: 10.1109/ISGT.2016.7781159.

[8] J. Tao and G. Michailidis, "A statistical framework for detecting electricity theft activities in smart grid distribution networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 1, pp. 205–216, 2020.

[9] J. Y. Kim, Y. M. Hwang, Y. G. Sun, I. Sim, D. I. Kim, and X. Wang, "Detection for non-technical loss by smart energy theft with intermediate monitor meter in smart grid," *IEEE Access*, vol. 7, pp. 129 043–129 053, 2019.

[10] R. Jiang, R. Lu, Y. Wang, J. Luo, C. Shen, and X. Shen, "Energy-theft detection issues for advanced metering infrastructure in smart grid," *Tsinghua Science and Technology*, vol. 19, no. 2, pp. 105–120, 2014.

[11] E. Benedict, *Losses in electric power systems*. Purdue University, School of Electrical Engineering, 1992.

[12] J. Mendiola, *Distribution system loss and anomaly detection using advance metering*, https://www.facebook.com/upeeei/videos/137596641133230 Accessed: 1-5-2021, 2020.

[13] *Republic act no. 7832 - an act penalizing the pilferage of electricity and theft of electric power transmission lines/materials, rationalizing system losses by phasing out pilferage losses as a component thereof and for other purposes*, http://www.erc.gov.ph/Files/Render/media/986_RA%207832%20Anti%20Pilferage%20Act%20of%201994.pdf Accessed: 1-5-2021, Republic of the Philippines, 1994.

[14] D. Rivera. (Feb. 2019). "Regulatory lag slows down Meralcoâs smart metering system," The Philippine Star, [Online]. Available: https://www.philstar.com/business/2019/02/01/1889764/regulatory-lag-slows-down-meralcos-smart-metering-system.

[15] M. M. Velasco. (Feb. 2019). "Meralco deployment of smart meters could raise its power rates," Manila Bulletin, [Online]. Available: https://business.mb.com.ph/2019/02/15/meralco-deployment-of-smart-meters-could-raise-its-power-rates/.

[16] A. M. Kosek and R. Czechowski, "The most frequent energy theft techniques and hazards in present power energy consumption," in *2016 Joint Workshop on Cyber- Physical Security and Resilience in Smart Grids (CPSR-SG)*, 2016, pp. 1–7. DOI: 10.1109/CPSRSG.2016.7684098.

[17] J. Nagi, A. M. Mohammad, K. S. Yap, S. K. Tiong, and S. K. Ahmed, "Non-technical loss analysis for detection of electricity theft using support vector machines," in *2008 IEEE 2nd International Power and Energy Conference*, 2008, pp. 907–912. DOI: 10.1109/PECON.2008.4762604.

[18] C.-w. Hsu, C.-c. Chang, and C.-J. Lin, "A practical guide to support vector classification chih-wei hsu, chih-chung chang, and chih-jen lin," Nov. 2003.

[19] G. M. Messinis and N. D. Hatziargyriou, "Review of non-technical loss detection methods," *Electric Power Systems Research*, vol. 158, pp. 250–266, 2018, ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2018.01.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779618300051.

[20] Z. Zheng, Y. Yang, X. Niu, H. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2018.

[21] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *Journal of Machine Learning Research*, vol. 1, 1Ã¢ÂÂ40, Jan. 2009. DOI: 10.1145/1577069.1577070.

[22] Y. Bengio, *Practical recommendations for gradient-based training of deep architectures*, 2012. arXiv: 1206.5533 [cs.LG].

[23] J. E. Mendiola and M. A. A. Pedrasa, "Detection of pilferage in an ami-enabled low-voltage network using energy reading anomalies," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, 2019, pp. 1–6. DOI: 10.1109/SGSMA.2019.8784464.

[24] S. M. Kamel A. Alboaouh, "Impact of rooftop photovoltaics on the distribution system," https://downloads.hindawi.com/journals/jre/2020/4831434.pdf, 2020.

[25] X. Yuan, M. Shi, and Z. Sun, "Research of electricity stealing identification method for distributed pv based on the least squares approach," in *2015 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*, 2015, pp. 2471–2474. DOI: 10.1109/DRPT.2015.7432661.

[26] M. Ismail, M. F. Shaaban, M. Naidu, and E. Serpedin, "Deep learning detection of electricity theft cyber-attacks in renewable distributed generation," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3428–3437, 2020. DOI: 10.1109/TSG.2020.2973681.

[27] E. L. Ratnam, S. R. Weller, C. M. Kellett, and A. T. Murray, "Residential load and rooftop pv generation: An australian distribution network dataset," *International Journal of Sustainable Energy*, vol. 36, no. 8, pp. 787–806, 2017. DOI: 10.1080/14786451.2015.1100196. eprint: https://doi.org/10.1080/14786451.2015.1100196. [Online]. Available: https://doi.org/10.1080/14786451.2015.1100196.

[28] M. Abraham, *Encyclopedia of sustainable technologies*. Elsevier, 2017.

[29] Ausgrid, *Australian residential load data 2011*, https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data Accessed: 1-5-2021, 2011.

[30] I. P. A. D. T. F. W. Group, *Pes test feeder*, https://site.ieee.org/pes-testfeeders/resources/.

[31] scikit learn.org, *Cross-validation: Evaluating estimator performance*, (accessed July 4, 2021). [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html.

[32] tutorials point, *Artificial neural network - basic concepts*, (accessed January 11, 2020). [Online]. Available: https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_basic_concepts.htm.

[33] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].

[34] M. GreÌgoire, O. GenevieÌve, and M. Klaus-Robert, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*. Springer, 2012.

[35] T. Schaul, S. Zhang, and Y. LeCun, *No more pesky learning rates*, 2013. arXiv: 1206.1106 [stat.ML].

[36] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, *Activation functions: Comparison of trends in practice and research for deep learning*, 2018. arXiv: 1811.03378 [cs.LG].

[37] S. Mukherjee, D. Ganguly, and K. Chatterjee, "A review of various techniques to detect non technical loss (ntl) of electricity," Mar. 2019.

[38] A. Maamar and K. benahmed, "Machine learning techniques for energy theft detection in ami," Jan. 2018, pp. 57–62. DOI: 10.1145/3178461.3178484.

[39] N. Seliya, T. M. Khoshgoftaar, and J. Van Hulse, "A study on the relationships of classifier performance metrics," in *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, 2009, pp. 59–66. DOI: 10.1109/ICTAI.2009.25.

[40] M. F. Uddin, "Addressing accuracy paradox using enhanched weighted performance metric in machine learning," in *2019 Sixth HCT Information Technology Trends (ITT)*, 2019, pp. 319–324. DOI: 10.1109/ITT48889.2019.9075071.

Acknowledgment

First of all we would like to thank our adviser, Adonis Tio, for supervising our undergraduate project. Thank you for guiding us in every step of the way, and for critiquing our work down to the smallest detail. Thank you for replying to our emails and queries, and for allowing us to consult with you when needed. Maraming salamat po sir!

To Ma'am Nicolette Arriola, our examiner, we thank you for the time that you have put in reviewing our study.

To Engr. Justin Mendiola, the author of the Anomaly Coefficient Calculation method, we thank you for allowing us to consult with you regarding your electricity theft detection algorithm.

To Engr. Ethel Luya, we thank you for entertaining our questions regarding electricity theft detection in general.

The authors would like to individually express their gratitude.

Carl Lavilla would like to thank the following:

Zild and Kel, thank you for being the best research partners. I cannot imagine myself working on this project with anyone else, your determination and efforts are unmatched. I will definitely (not) miss our debates and sleepless nights. Thank you for being my comrades, even outside schoolwork. There is a long road ahead of us, and I am sure that whatever path you two will choose, you will be the best at it.

To my friends from EEE and UP Circuit, thank you for the support and guidance throughout the years. Special mention EE bros: Drew, Gab, Gords, Roi, Sean, Ram, Drei, sagot ko na susunod nating punta sa Hardin.

To my HS friends that are like brothers to me: Agui, Leo, Tim, Jason, Dum, Bon, Kirk, Miel, Gats, thank you for being there for me when I needed a break.

To my mother, Mariquita Lavilla, thank you for your everlasting love and support in everything that I do. Thank you for always believing in me. To my father, Claro Lavilla, thank you for all the sacrifices that you have made for us. You are my inspiration and favorite teacher. Mom and dad, thank you for all the prayers and words of encouragement, I could have never done it without you. To my sisters: Peebs and Cia, thank you for your understanding and positive energy. I love you all.

To God be the glory.

Michael Osorio would like to thank the following for everything throughout his college life:

To my group mates, Carl and Zild, this project wouldn't be the same without you. To Carl, thank you for accepting my offer of being partners. I know I always voice it out as a solution to help you in your project but I guess it really was a call for help for not wanting to be alone. To Zild, thank you for tagging along. You're the last piece that we needed to get this boat rocking and you are more than what we hoped for. Thank you to both of you and I'll always be ready to help you in your future endeavors.

To my OG organization in UP, UP Circuit. I loved you, love you, and will always love you. Thank you for letting me meet the people that will make my College life not easy, but worth it. Thank you for giving me the opportunity to be served, learn, and then serve others. You've let me fail and then succeed, lose and then win, but more importantly, receive and then give. And I wasn't alone on this journey. You gave me a lot of friends that is more than I could have ever hoped for. To Laarni, Ryndgel, John, Jho, Renz, CA, Marvin, Pat, and Toto, thank you for giving me the best introduction year to UP. To the strongest division, Internal Affairs Division, I love you guys all the best and you will always have my support. Special shoutout to Mama Jer, for being the best Inte Mentor and to Alex, Jacy, Nadz, Joph, Bags, Ara, JC, Gordon, Allysa, Karl, Destu, Drew, Ram, Marianne, Malkiy, Bon, Bea and all the new inte mems for making my VP role easier. To the new inte generation, you got this. To the best ExeBoard Members, Jonathan, Elysse, Efraim, Myka, Eira, Stan, and CA, thank you for giving me the opportunity to give my best to the organization. And lastly, a special shoutout to my partner Janet for giving me the opportunity to lead my dream role, Engg Week Head. Many thanks to everyone that helped us.

To my family, thank you for not pressuring me and for giving me the best support system that I can have. To my mama, Jannie Anne Creus, thank you for making sure that there's a roof over my head and food for my big tummy. Thank you for the life lessons that I wouldn't need to experience to learn from. To my papa, Edwin Doctor, thank you

for spoiling me in the best way possible. Thank you for being outgoing and for letting me learn lessons through my own personal experience. To both of you, thank you for providing the ultimate support but still letting me experience life as free as it could be experienced.

To Myka Maala, thank you for everything. Thank you for the stay overs to help me study, to the eat outs to let me relax, and the night outs to help me unwind. Thank you for being the best at motivating me and also the best at understanding me. I know there's been a lot of ups and downs but thank you for staying with me through the end of this ligawan session. Yes, end of this ligawan session because finally, after n years and x months of on and offs, I can now pop the question. Will you be my girlfriend?

Zildjian Joshua Restituto would like to thank the following for their continuous support throughout this project:

To my research groupmates, Michael Osorio and Carl Lavilla, thank you for adopting me at the last minute and making our last semesters fun and exciting. This project would not have been finished if not for your teamwork and determination. Know that you can always count on me– I am only one Discord-call away.

To my second family, the Alpha Phi Beta Fraternity, thank you for providing me a home away from home throughout my college years. I would not be the person I am today without the Fraternity. To my closest brods: Jon, Neil, Sol, Tim, Alvz, Paco, Rosh, and Gab, thank you for the unwavering support.

To my HS best friends: Rinz, Prince, and Remcel, I thank you for motivating me. You guys are my support system since day one.

To Hanna Bulacan, thank you for never giving up on me. From sending coffee and goodies to waking me up, your support gave me the energy and determination to keep going, one day at a time. You inspire me every day. I love you.

To my family, thank you for your understanding whenever I cannot fulfill my duties at home during the project. To my beautiful sisters, Kassandra Restituto and Alessandra Restituto, thank you for being my main support system and for motivating me everyday. Every day I aspire to be a better person because of you two. To my father, Danilo Restituto, thank you for providing the love and care even if we are hundreds of miles apart. To my superhero mother, Anna Cantos, thank you for tirelessly supporting my interests and for never doubting me. You are my forever role model. I know that I can never repay all the sacrifices you made but know that I will always do my best to make you proud. You guys are literally the reason that I exist and for that I am eternally grateful. I love you all so much. Special mention to my boy Pugsley, who made my every day better especially during the pandemic despite his sleep-depriving barks at night.

Carl Samuel R. Lavilla
2015-05369
Bachelor of Science in Electrical Engineering

Michael E. Osorio
2015-05350
Bachelor of Science in Electronics and Communications Engineering

Zildjian Joshua C. Restituto
2015-00863
Bachelor of Science in Electrical Engineering