Proceedings Paper:
Alromih, A., Clark, J.A. and Gope, P. (2021) Electricity theft detection in the presence of prosumers using a cluster-based multi-feature detection model. In: 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 25-28 Oct 2021, Aachen, Germany. Institute of Electrical and Electronics Engineers. ISBN: 9781665430449.

https://doi.org/10.1109/smartgridcomm51999.2021.9632322

This work has been accepted for publication at the IEEE SmartGridComm2021 Conference

Electricity Theft Detection in the Presence of Prosumers Using a Cluster-based Multi-feature Detection Model

Arwa Alromih *†, John A. Clark* and Prosanta Gope*
*Department of Computer Science, The University of Sheffield, Sheffield, UK
{asmalromih1, john.clark, p.gope}@sheffield.ac.uk
†Information Systems Department, King Saud University, Riyadh, Saudi Arabia

***Abstract*—Data driven approaches have been widely employed in recent years to detect electricity thefts. Although many techniques have been proposed in the literature, they mainly focus on electricity thefts by consumers of power from the grid. Existing studies do not consider electricity thefts by *prosumers*, who act as both supplier and consumer in the energy system. This is of great importance as inaccurate reports of prosumers' behaviours can disturb power system operation. Here, the paper examines the role prosumers may play in subverting the energy system and propose a novel means of detecting such malfeasance. Specifically, this work introduces a *new* electricity theft attack scenarios called *balance attacks*, where an attacker concurrently modifies his readings along with neighbouring meters in an attempt to balance the total aggregated reading. Such attacks can be difficult to detect by existing solutions that reach detection decisions based on aggregated readings.A novel electricity theft detector is proposed that can detect thefts in the presence of prosumers. Current approaches use either a single model for all users across the system or else a model for each user. Here, a half-way house approach is adopted where a cluster-based detection model is used. In each cluster, the power time series for a user is decomposed into trend, cyclical and residual components. Residual data, along with different features from multiple data sources, are fed in an ML classification algorithm to detect anomalous readings. Simulations have been conducted using a newly generated dataset and results have shown that the proposed model can detect electricity theft with high detection and low error rates. The results also shows that the proposed model can detect thefts with great accuracy from new users.**

***Index Terms*—Advanced metering infrastructure (AMI), energy theft, smart grid, electricity theft, prosumer.**

I. Introduction

Energy plays an essential role in our daily lives. Integrating the existing energy distribution networks with information and communications technology (ICT) has introduced the concept of advanced metering infrastructure (AMI), where electricity providers, utility companies and users actively communicate with each other for offering a reliable and cost-effective demand-response management [1]. In AMI, a set of smart meters (SMs) are deployed at users' premises to send fine-grained measurements of consumed and generated electricity to the control centers. These meter data are the main inputs in critical decision-making processes related to energy efficiency, optimization, and operational reliability. Therefore, the integrity of these data must be guaranteed. False data injection (FDI) attacks can be one of the most critical and serious attacks launched against data integrity in energy management systems. Faults in data can negatively impact the operations of the AMI, resulting in blackouts [2]. FDI attacks are maliciously engineered data corruption attacks where an attacker manipulates reported sensor measurements to achieve their goals, most typically subverting the control of the system itself or its underpinning accounting system.

Recently, the incorporation of distributed energy resources (DERs) in a user's premises, allowing a user to generate, store and supply electricity, has received significant attention. Here, the stakeholder is generally referred to as a "prosumer" (producer-consumer). Considering the role of prosumers is important as their number is increasing rapidly; according to the European Renewable Energies Federation [3], the UK in 2015 had almost 1 million prosumers and will likely reach 24 million by 2050. Prosumers' thefts can be carried out through the manipulation of consumption and generation data. The manipulation of generation readings can be more dangerous as they can affect electricity generation, the energy markets and the demand-response models. Therefore, the detection of prosumers attacks is of a great importance.

Traditionally, electricity theft was detected by on-site inspections of meters in order to identify faulty or tapped ones. This is expensive in both time and cost [4]. With the introduction of smart meters, more sophisticated digital methods were introduced to detect electricity theft. Some of these ways are addressed below.

A. Related Work and Limitations

Several approaches have been proposed in the literature to defend against electricity theft, e.g. state-estimation, game-theory based and machine learning approaches. Machine learning (ML) is increasingly being used because of its ability to be scaled to large systems and its low computational costs [1]. Supervised, semi-supervised, and unsupervised learning have all been proposed for detecting electricity theft. The features of several works are summarized in Table I.

In [5], the authors used a convolutional neural network (CNN) to automate feature extraction, and a CNN-based long short-term memory (LSTM) model to detect energy thieves. Their work has achieved a plausible accuracy rate of 89% but a lower detection rate (recall) of around 87%. Another technique

TABLE I
COMPARISON OF ML-BASED DETECTION RESEARCH WORK

Detection Approach	ML Method[a]	Features[b]				
		F1	F2	F3	F4	F5
Hasan et al. [5]	CNN-LSTM	✗	✓	✗	N/A	✗
Zanetti et al. [6]	FCM	✗	✓	✓	✗	✗
Hu et al. [7]	DSN + DAE	✗	✓	✓	✓	✗
Gunturi and Sarkar [8]	Ensmble ML	✗	✓	✓	N/A	✗
Yan and Wen [9]	XGBoost	✗	✓	✗	N/A	✗
Our approach	Various MLs	✓	✓	✓	✓	✓

[a] FCM: Fuzzy C-Means; CNN-LSTM: Convolutional Neural Network - Long Short Term Memory; DSN: Deep Siamese Network; DAE: Denoising Autoencoder; XGBoost: Extreme Gradient Boosting;
[b] **F1**: Ability to detect prosumers' thefts; **F2**: Ability to pinpoint a thief; **F3**: Ability to pinpoint time of theft; **F4**: No requirement for historical data; **F5**: Usage of multi-source data.

in [7] has shown a better detection rate where a semi-supervised technique was used to train the detection model. In the semi-supervised training, labelled samples and unlabelled samples are used to train a deep Siamese network (DSN) model and de-noising auto-encoder (DAE) respectively. On the other hand. Zanetti et al. [6] have proposed an unsupervised clustering algorithm to detect energy thefts using short-lived consumption patterns. These patterns represent the profile of the consumer over a short period of time. In their model, the authors tested the detection accuracy with different pattern durations (ranging from 1 day to 3 weeks). The results showed that increasing the duration time does not necessarily improve the detection of thefts. Gunturi and Sarkar [8] introduced an energy theft detection model for AMI based on ensemble ML techniques. The idea of ensemble ML models is to combine multiple ML approaches into one predictive model to boost the detection rate and lower the error rate. In their study, the authors found that a bagging-type ensemble ML approach, which combines the results of independent MLs parallel by taking the average, performs better than a boosting one. Another recent study [9] have used extreme gradient boosting (XGBoost) which is a scalable implementation of decision tree boosting system. The study showed that the XGBoost model is robust when the dataset is imbalanced.

No existing research has studied the impact of prosumers' thefts (represented as F1 in Table I). Prosumers are different from traditional consumers as they do not only use energy but also generate and store or transfer surplus energy to the grid [10]. Prosumers can affect electricity generation, the energy markets and the demand-response models. To manage prosumers, it is critical to understand their generation and consumption behaviours [10]. The analysis of all factors of prosumer behaviours helps to build and plan for the proper balance of energy demand and supply. A malicious prosumer who reports falsified data regarding his/her generation and consumption can disrupt the supply of energy to a region, cause grid instability or deny energy access to other users in that area [11]. Moreover, some existing energy theft detection research does not identify *which* user is the thief (represented as F2 in Table I) and many other detection methods classify each user as either thief or honest but do not identify the time

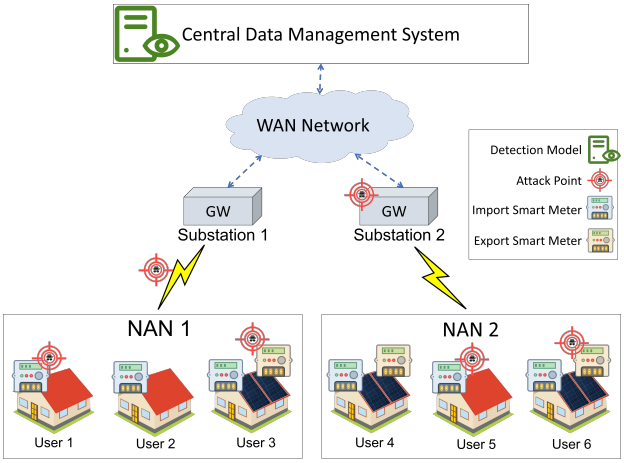

Fig. 1. System architecture showing the possible attack points

of theft (represented as F3 in Table I). Additionally, most recent studies have not availed themselves of data features from different sources (represented as F5 in Table I). Machine learning approaches usually consider a single electrical feature (consumed power), while smart meters report more than ten different electrical parameters. This abundance of unused data is an opportunity.

B. Our Contributions

This paper addresses all the above limitations. The specific contributions of our paper are:

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

The rest of this paper is organized as follows: Sections II and III provide the system architectural model and the adversarial model. Section IV describes how the proposed detection system is designed. Section V details the experimental setup and results. Finally, Section VI gives concluding remarks and directions for future work.

II. SYSTEM ARCHITECTURE

Figure 1 shows the system architecture for an AMI. It consists of three major entities: a set of users, a set of substation gateways (GWs), and the central data management system (CDMS). A user can be either a consumer (who consumes

electricity from the grid) or a prosumer (who both supplies and consumes electricity to/from the grid). Prosumers generate their own electricity from distributed energy resources (DER) such as solar panels or wind turbines. A consumer, i, is equipped with only one meter called an import SM (ISM_i) which calculates the amount of electricity consumed. A prosumer j, who has a dedicated DER, is equipped with two types of smart meter: an import SM (ISM_j) and and export SM (ESM_j). Export SMs calculate the electricity supplied from the prosumer's DER to the grid. Both types of SMs collect and report users' consumption and generation data to the GW on a regular basis (say, every 15 minutes). A substation gateway is located in a neighbourhood area network (NAN) that serves a group of users. The GW is in charge of collecting the SMs' data and sending these reports to the CDMS. Detection of abnormalities in either consumption or generation reports is carried out at the CDMS.

III. Adversary Model

Electricity theft can be carried out by manipulating the electricity reading reports. In the considered adversary model, a malicious user is allowed to change their meter readings to pay a lower consumption bill or get paid for electricity that they did not generate. An adversary can manipulate consumption and generation readings at any attack point as shown in Fig.1 (the system architecture). The adversary can inject false measurements by physically manipulating the configuration of a smart meter or by attacking the communication channels. Therefore, our adversary model considers two types of adversaries:

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

Both external and internal adversaries can modify both the meter readings of import SM (ISM) or export SM (ESM) using different attack scenarios as listed in Table II. The adversary model considers eight different attack scenarios that can be launched by the two adversaries mentioned before. The first four of these attacks have been developed with the help of the widely used mathematical model defined in [12]. The additional four scenarios have considered attacks where one reported consumption is maliciously *increased* to balance a malicious decrease in another. This model incorporates such attacks and refer to them as *balance attacks*. These attacks are assumed to be launched by either a single attacker or in a collaborative manner using collusive attacks. All attacks considered in this model are described below and are summarized in Table II.

- In attack scenarios #1 and #2 user i (either consumer or prosumer) decreases his import smart meter readings ISM_i by a constant value l or a constant percentage k.

TABLE II
Overview of Attack Scenarios

Attack Scenario		Modification
	Attack #1	$ISM'_i = ISM_i - l$
	Attack #2	$ISM'_i = ISM_i \times (1 - \frac{k}{100})$
	Attack #3	$ESM'_i = ESM_i + l$
	Attack #4	$ESM'_i = ESM_i \times (1 + \frac{k}{100})$
Balance Attacks	Attack #5	$ISM'_i = ISM_i - l \quad and$ $ISM'_j = ISM_j + l$
	Attack #6	$ISM'_i = ISM_i \times (1 - \frac{k}{100}) \quad and$ $ISM'_j = ISM_j + (ISM_i \times \frac{k}{100})$
	Attack #7	$ESM'_i = ESM_i + l \quad and$ $ESM'_j = ESM_j - l$
	Attack #8	$ESM'_i = ESM_i \times (1 + \frac{k}{100}) \quad and$ $ESM'_j = ESM_j - (ESM_i \times \frac{k}{100})$

- In attack scenarios #3 and #4 prosumer i increases his export smart meter readings ESM_i by a constant value l or a constant percentage k.
- In attack scenarios #5 and #6 user i (either consumer or prosumer) decreases his import smart meter readings ISM_i by a constant value l or a constant percentage k and adds the same value to the meter report of some other user j in the same NAN.
- In attack scenarios #7 and #8 prosumer i increases his export smart meter readings ESM_i by a constant value l or a constant percentage k and decreases the same value from the meter report of some other prosumer j in the same NAN.

IV. Proposed Detection Model

Our detection approach seeks to distinguish sets of theft points from sets of other points. Since thefts are expected to be comparatively rare, this essentially means that they are identified as a form of *outlier*. Our approach is thus one of anomaly detection. The results suggest that this approach is well-founded. There may be other sources of outliers; this is always possible with an anomaly detection approach. Fig.2 shows the three phases of our proposed detection approach. The phases are described below.

A. User Clustering

Extant research uses either a generalised model (where one honest reference model is built with data from all users) or else employs a user-specific approach (where a model is created specifically for each user using that user's data) [13]. Generalised models can exhibit low accuracy whilst user-specific models encounter significant scaling issues. Our approach offers a half-way house: it clusters users and develops a reference model for each cluster. Users are clustered based on their geographical location and user residence characteristics. Users who share the same geographical location and residence physical characteristics are likely to have a similar pattern of consumption and generation. According to Eurostat [14],

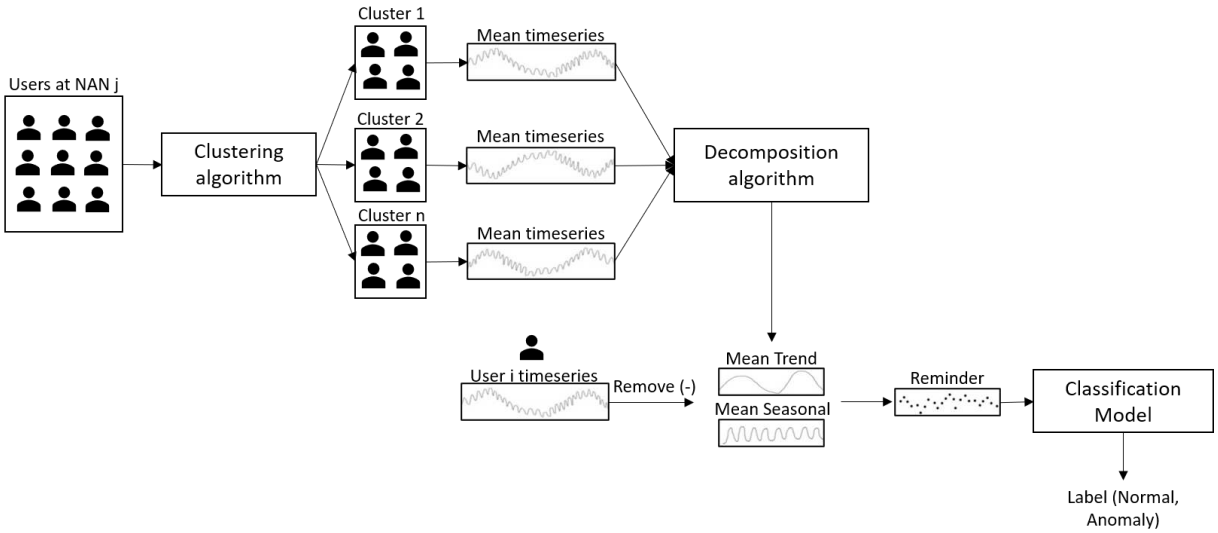

Fig. 2. Overview of The Detection System

people in the same neighbourhood are more likely to have similar incomes, which in turn, affects the physical characteristics of their building and the types of equipment and appliances. Therefore, their consumption and generation patterns will typically be similar, whilst users in different clusters can have different usage and generation profiles. In this phase, the 14 static features (reported in Table III) are normalized using the Standardscalar technique and then *agglomerative clustering* is used on the normalized data to partition users with the number of clusters in each NAN based on minimizing the total within-cluster sum of square (WSS) (Elbow method). The data from each cluster is then processed individually in the next phase. This phase is executed only once, either at the beginning of the system deployment or after the registration of new users.

B. Timeseries Decomposition

As discussed in Section IV, the proposed detection method is one of anomaly detection where theft points are regarded as anomalies. However, data taken at different times may have structural differences, e.g. due to the season. To place all data points on a comparable footing, we remove certain systematic elements, specifically the trend and seasonal effects, leaving so called *residuals*. This is usually referred to as timeseries decomposition. A timeseries data Y at time t is composed of three components: a trend component T_t, a seasonal component S_t and a residual (remainder) component R_t. These components are either added or multiplied together to form the original signal.

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

To automatically decompose a timeseries into its components, different methods have been proposed, such as, seasonal-trend decomposition using regression (STR) [15], singular spectrum analysis (SSA) [16], and decomposition of time series by Loess (STL) [17]. In this paper, the additive STL is used to decompose the users consumption and generation series. As compared to STR and SSA, the period of the seasonality

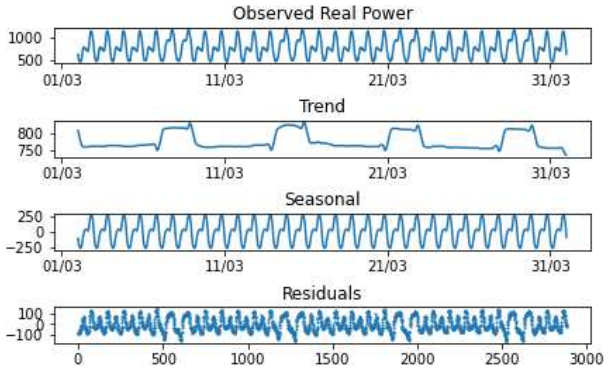

Fig. 3. Decomposition of Consumed Power for a Cluster

component 'cyclical variation' in STL can be interpreted flexibly according to need. Indeed, in our experiments over one month's data, a daily cycle is chosen in place of a seasonal one. For each cluster, the average consumption/generation of all users is computed and then decomposed to obtain the cyclical (here daily) and trend components (see Fig.3). For each user, those components are removed from his/her data to obtain the residuals. Obtaining the residuals from removing the cluster's trend and daily components is found to create more distance between normal and anomalous data points than removing all users' trend and daily components. This increase in distance, as shown in Fig.4, creates a separation between normal and anomalous data.

C. Classification

In the final phase, each data point which is a vector of 15 features (both dynamic and weather features reported in Table III) is classified as either anomaly or normal data. In the training phase, a balanced dataset of equal normal and anomalous data points is used. The dataset involves multiple features along with the residuals obtained from the previous phase. As each data point consists of a vector of features with different value ranges, these features are first normalized using

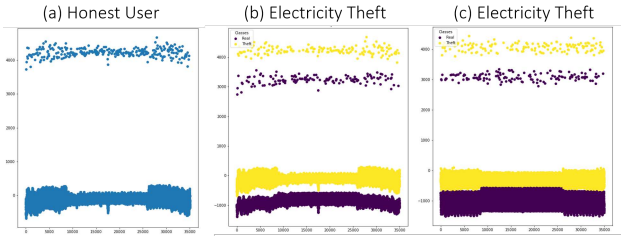

Fig. 4. Residuals of (a) Honest User, (b) Electricity Thief after removing the trend and daily components of *all users*, and (c) Electricity Thief after removing the trend and daily components of *cluster's users*. Purple points indicate normal data points while yellow ones are thefts.

a StandardScalar [18]. Different machine learning algorithms are then applied on this normalized data to make the decision. After training the model, it is used to detect data thefts in unseen data points. At the end of this phase, the system will have a single detection model for each cluster of users.

V. Performance Evaluation

In this section, the feasibility of the detection system is examined in terms of accuracy, recall (detection rate), precision and error rate.

A. Experimental Setup

1) Dataset Generation: In order to study the electricity theft scenarios done by both consumers and prosumers, a dataset that include electricity consumption of users from both types is needed. Most current literature that studies electricity thefts use one of these two public electricity consumption datasets. The first and most widely used dataset is the one released by the State Grid Corporation of China (SGCC) (the largest electricity utility in China) [19]. This dataset is the first to include realistic labeled data, where each user is labeled as honest or a thief. A downside to this dataset is that the consumption is reported only once a day which makes it difficult to identify the exact time of theft [20]. The second widely used dataset is the Irish Commission for Energy Regulation (CER) Smart Metering load profiles [21] which contains the consumption data of over 5000 residential and enterprise users for a duration of 500 days. However, this dataset contains only honest profiles and reports only the consumed real power at a half hourly rate. The two datasets also lack contextual data that might affect the consumption of a user such as the floor area of the residency, location and weather conditions.

Due to the absence of a public database containing both consumers and prosumers, a new dataset has been generated using the "GridLab-D" simulation tool [22]. The taxonomy distribution feeder, R1-12.47-2, which was developed by Pacific Northwest National Laboratory (PNNL) [23], was used to produce a detailed distribution feeder model in GridLab-D format that could be used to generate the dataset for our work. This distribution feeder represents a moderately populated suburban and rural area composed of 1594 residential users with varying loads and physical properties, where 49 of those users are prosumers with solar panels. Our dataset not only contains consumption and generation profiles of both consumers and prosumers, but it also reports multiple electrical parameters every 15 minutes. It also contains weather conditions and users' static residence characteristics. The script provided by PNNL has been modified to allow the reporting of the weather and dynamic features listed in Table III every 15 minutes for every user. Note that the proposed model is adjustable to any reporting frequency, and can be applied to any dataset providing that it has some static parameters to cluster the users.

TABLE III
Features of The Dataset

Static Parameters	Dynamic Parameters	Weather Parameters
Floor area	Consumed real power	Temperature (Dry-Bulb)
Number of stories	Voltage	Pressure
Ceiling height	Real energy	Humidity
Roof's R-value	Reactive energy	Total sky cover
Wall's R-value	Reactive power	Extraterrestrial radiation
Floor's R-value	Current	Wind speed
Door's R-value	Apparent power	Wind direction
Number of glazing layers	Generated real power	
Glass type		
Glazing treatment		
Window frame type		
Heating system		
Cooling system		
Solar panel size		

2) Attack Modes Simulation: The readings in our dataset were modified as they only contain honest (real) readings in order to define the set of theft scenarios that were considered in the adversary model. Several existing literature have followed the same design where data theft scenarios are synthetically added to a dataset, in order to use them for training and evaluating their detection model. Nine different datasets were created: one for every attack scenario and one dataset with all 8 attacks combined. In the experiments, the values of l and k, defined in the attack scenarios in Table II were set to 500 and 40 respectively. For the sake of research reproducibility, the original dataset has been published in our Github repository[1].

3) Simulation Environment: The proposed detection model is tested using several benchmark ML algorithms. These ML algorithms were trained and tested using scikit-learn[18] in the Anaconda3 environment using Python. For each ML algorithm, the default hyper-parameters provided by scikit-learn were used. All results reported are the average of validating the model using 10-fold cross-validation. In the first phase, *clustering* phase, the set of static parameters shown in Table III are used, where as the remaining set of parameters, along with the residuals from the second phase, are used in the *classification* phase.

4) Evaluation Metrics: Several metrics are used to evaluate the proposed system: accuracy, recall (detection rate), precision and error rate. Other metrics such as F-score can be easily calculated from the reported metrics. Our motivation is to obtain high accuracy and recall (detection rate) with a low error rate. In this paper, theft data points are denoted as positive class and benign data points as negative class. A confusion matrix is used in order to evaluate the performance of the electricity theft detection model where True Positive

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

TABLE IV
EXPERIMENTAL RESULTS OF THE PROPOSED MODEL UNDER DIFFERENT ATTACKS

Attack Scenario	Accuracy[a]						Recall (DR)						Precision						Error Rate					
	DT	KNN	LR	NB	NN	SVM	DT	KNN	LR	NB	NN	SVM	DT	KNN	LR	NB	NN	SVM	DT	KNN	LR	NB	NN	SVM
Attack #1	0.996	0.941	0.997	0.783	**0.999**	0.992	0.995	0.973	0.994	0.912	**1.000**	0.994	0.995	0.920	**1.000**	0.758	0.998	0.990	0.004	0.059	0.003	0.217	**0.001**	0.008
Attack #2	0.989	0.887	0.994	0.647	**0.998**	0.993	0.991	0.961	0.999	0.950	**1.000**	0.999	0.988	0.848	0.989	0.617	**0.996**	0.988	0.011	0.113	0.006	0.353	**0.002**	0.007
Attack #3	0.983	0.896	0.527	0.637	**0.997**	0.888	0.980	0.913	0.481	0.429	**0.995**	0.839	0.985	0.886	0.549	0.700	**0.999**	0.921	0.017	0.104	0.473	0.363	**0.003**	0.112
Attack #4	0.964	0.747	0.501	0.555	**0.995**	0.726	0.958	0.803	0.474	0.436	**0.991**	0.613	0.969	0.727	0.499	0.605	**0.998**	0.781	0.036	0.253	0.499	0.445	**0.005**	0.274
Attack #5	**0.983**	0.963	0.936	0.879	0.978	0.970	0.984	**0.987**	0.926	0.840	0.986	0.968	**0.984**	0.943	0.948	0.915	0.971	0.973	**0.017**	0.037	0.064	0.121	0.022	0.030
Attack #6	**0.939**	0.867	0.841	0.804	0.911	0.892	**0.946**	0.937	0.790	0.725	0.904	0.827	0.933	0.822	0.883	0.866	0.920	**0.954**	**0.061**	0.133	0.159	0.196	0.089	0.108
Attack #7	0.966	0.932	0.499	0.821	**0.976**	0.923	0.976	0.949	0.397	0.755	**0.980**	0.899	0.954	0.916	0.508	0.865	**0.974**	0.943	0.034	0.068	0.501	0.179	**0.024**	0.077
Attack #8	**0.920**	0.840	0.501	0.775	0.886	0.844	**0.925**	0.892	0.382	0.659	0.876	0.786	**0.915**	0.803	0.485	0.848	0.894	0.887	**0.080**	0.160	0.499	0.225	0.114	0.156
All Attacks	0.884	0.802	0.742	0.571	**0.935**	0.797	0.871	0.831	0.695	0.257	**0.938**	0.658	0.908	0.789	0.788	0.701	**0.936**	0.912	0.116	0.198	0.258	0.429	**0.065**	0.203

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

(TP) denotes the number of correctly identified attacks and False Positive (FP) is the number of normal records incorrectly identified as attacks. The True Negative (TN) is the number of normal records that are correctly identified as normal and the False Negative (FN) denotes the number of attack records that are incorrectly identified as normal. The evaluation metrics have the following notation:

- Accuracy: how many samples were classified correctly out of the total sample population.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{(ii)}$$

- Detection Rate: the fraction of actual attacks that are detected. This measure is also called detection rate (DR).

$$DetectionRate = \frac{TP}{TP + FN} \quad \text{(iii)}$$

- Precision: the number of correctly detected attacks divided by the number of total detections.

$$Precision = \frac{TP}{TP + FP} \quad \text{(iv)}$$

- Error Rate: The ratio of incorrect predictions (number of false alarms generated for normal samples + number of attacks missed) over the total sample population

$$ErrorRate = \frac{FP + FN}{TP + TN + FP + TN} \quad \text{(v)}$$

B. Results and Discussion

The detection system was evaluated in terms of:

- Impact of different types of attacks.
- Detecting theft from new users.
- Impact of changing the percentage of thieves among the users.

1) Impact of Different Attacks: The overall detection performance was tested for each attack scenario discussed in Section III and also for attacks in the combined dataset. Table IV shows the accuracy, recall (detection rate), precision and error rate of these different attack scenarios. As indicated above, the results reported are the average of a 10-fold cross validation over a balanced dataset. Several ML techniques have been used for the classification phase.

The results in Table IV show that our detection model has a good performance in detecting all attack types. From Table IV, attacks #1 #2, #3 and #4 are detected with a detection rate of 100%, 100%, 99.5% and 99.1% respectively using a neural network ML model. Attacks #5, #6, #7 and #8 are detected with a detection rate of above 92%. In the combined dataset $AllAttacks$, the detection model can detect any attack type with a detection rate of 93.8%. These results show that the proposed model can detect different attacks with high detection probability. It can also be seen that the detection in attacks #5, #6, #7 and #8 is slightly lower than the other types. Balance attacks seem a little more difficult, perhaps an intrinsic property of zero overall theft.

2) Impact of Thefts From New Users: Here, the proposed detection model is evaluated in terms of detecting thefts from new users. First, the classifier is trained using the combined dataset $AllAttacks$ that includes samples of all attacks types. After that, a test dataset of users that have not been included during the training phase is used to evaluate the proposed detection model. Table V shows how well the detection model works. It can be observed that the best performance in terms of accuracy, recall, precision and error rate was given by the neural network classifier. Our model can detect thefts from new users without the need for historical data with a detection rate of 93.2% and only 7.1% error rate.

3) Impact of Different Percentage of Thieves: This setting analyses the effect of the percentage of thieves that exists in a singe cluster. As this is an important factor to take into consideration which can show how well the detection algorithm works in cases of low numbers of thieves. The experiments were conducted using the combined dataset $AllAttacks$ which contains all attack types. In this setting, the model was trained using 10-fold cross validation and then tested using a completely unseen and unbalanced dataset. Table

TABLE V
PERFORMANCE OF THE DETECTION ON THEFTS FROM NEW USERS

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

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

TABLE VI
PERFORMANCE OF THE DETECTION MODEL UNDER DIFFERENT PERCENTAGE OF THIEVES

Percentage of Theft	Accuracy[a]						Recall (DR)						Precision						Error Rate					
	DT	KNN	LR	NB	NN	SVM	DT	KNN	LR	NB	NN	SVM	DT	KNN	LR	NB	NN	SVM	DT	KNN	LR	NB	NN	SVM
2%	0.973	0.852	0.778	0.903	**0.987**	0.973	0.995	**1.000**	**1.000**	0.125	**1.000**	**1.000**	0.427	0.122	0.085	0.032	**0.617**	0.430	0.027	0.148	0.222	0.097	**0.013**	0.027
5%	0.977	0.859	0.788	0.821	**0.988**	0.978	0.994	0.990	0.992	0.310	**0.996**	0.985	0.695	0.266	0.194	0.100	**0.812**	0.706	0.023	0.141	0.212	0.179	**0.012**	0.022
10%	0.971	0.863	0.777	0.720	**0.988**	0.975	0.995	0.979	0.956	0.321	**0.999**	0.955	0.780	0.427	0.310	0.135	**0.895**	0.828	0.029	0.137	0.223	0.280	**0.012**	0.025
20%	0.972	0.879	0.761	0.765	**0.989**	0.934	0.994	0.971	0.795	0.217	**1.000**	0.788	0.883	0.635	0.453	0.376	**0.950**	0.877	0.028	0.121	0.239	0.235	**0.011**	0.066

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

VI shows the results where the percentage of thieves in a cluster was randomly changed to range from 1% to 20%. The results indicates that our method actually achieves an excellent detection rate and minimal error rates with varying percentages of theft. Our model shows an average detection rate of above 97% using decision tree, KNN and neural network classifiers.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposed a data-driven electricity theft detection mechanism in the presence of prosumers attacks. The detection approach is designed to detect different electricity theft attacks from both consumers and prosumers by analysing SMs reading reports in a cluster-based manner. Moreover, a new attack scenario was introduced *balance attacks* where attackers try to conceal their theft by balancing the total net of consumed or generated power. Simulations are done using a generated dataset that comprise of generation and consumption profiles of both prosumers and consumers along with data from multiple data sources. Results show that the proposed model has a high detection performance for each type of attack and an overall 93% detection rate. The detection model is also tested when different percentage of thieves in a cluster. Results show that the proposed method achieves good detection rate when data tested is imbalanced. While smart meters offer some clear benefits, fine-grained measurements of household energy consumption trigger serious privacy concerns. In this regard, fine-grained smart meter data may reveal a user's presence/absence in his/her house, which electrical appliances they are using at any moment, or even their daily habits at home. Therefore, privacy-enhanced ML approaches are currently being investigated for energy systems.

REFERENCES

[1] E. Hossain, I. Khan, F. Un-Noor, S. S. Sikander, and M. S. H. Sunny, "Application of big data and machine learning in smart grid, and associated security concerns: A review," *IEEE Access*, vol. 7, pp. 13 960–13 988, 2019.

[2] M. G. Kallitsis, S. Bhattacharya, and G. Michailidis, "Detection of false data injection attacks in smart grids based on forecasts," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2018, pp. 1–7.

[3] I. Clover, "UK could be home to 24 million clean energy prosumers by 2050," *PV Magazine*, September 2016. [Online]. Available: https://www.pv-magazine.com/2016/09/27/uk-could-be-home-to-24-million-clean-energy-prosumers-by-2050-says-report-100026268

[4] Y. Gao, B. Foggo, and N. Yu, "A physically inspired data-driven model for electricity theft detection with smart meter data," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5076–5088, 2019.

[5] M. Hasan, R. N. Toma, A.-A. Nahid, M. Islam, J.-M. Kim *et al.*, "Electricity theft detection in smart grid systems: A cnn-lstm based approach," *Energies*, vol. 12, no. 17, p. 3310, 2019.

[6] M. Zanetti, E. Jamhour, M. Pellenz, M. Penna, V. Zambenedetti, and I. Chueiri, "A tunable fraud detection system for advanced metering infrastructure using short-lived patterns," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 830–840, 2017.

[7] T. Hu, Q. Guo, X. Shen, H. Sun, R. Wu, and H. Xi, "Utilizing unlabeled data to detect electricity fraud in ami: A semisupervised deep learning approach," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3287–3299, 2019.

[8] S. K. Gunturi and D. Sarkar, "Ensemble machine learning models for the detection of energy theft," *Electric Power Systems Research*, vol. 192, p. 106904, 2021.

[9] Z. Yan and H. Wen, "Electricity theft detection base on extreme gradient boosting in ami," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[10] E. Espe, V. Potdar, and E. Chang, "Prosumer communities and relationships in smart grids: A literature review, evolution and future directions," *Energies*, vol. 11, no. 10, p. 2528, 2018.

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

[12] P. Jokar, N. Arianpoo, and V. C. Leung, "Electricity theft detection in ami using customers' consumption patterns," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 216–226, 2015.

[13] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Robust electricity theft detection against data poisoning attacks in smart grids," *IEEE Transactions on Smart Grid*, 2020.

[14] Eurostat, "Manual for statistics on energy consumption in households," 2013. [Online]. Available: https://ec.europa.eu/eurostat/documents/3859598/5935825/KS-GQ-13-003-EN.PDF/baa96509-3f4b-4c7a-94dd-feb1a31c7291

[15] A. Dokumentov and R. J. Hyndman, "Str: A seasonal-trend decomposition procedure based on regression," *arXiv preprint arXiv:2009.05894*, 2020.

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

[17] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition," *Journal of official statistics*, vol. 6, no. 1, pp. 3–73, 1990.

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

[19] Z. Zheng, Y. Yang, X. Niu, H.-N. 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, April 2018.

[20] M. Adil, N. Javaid, U. Qasim, I. Ullah, M. Shafiq, and J.-G. Choi, "Lstm and bat-based rusboost approach for electricity theft detection," *Applied Sciences*, vol. 10, no. 12, p. 4378, 2020.

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

[22] US Department of Energy, "GridLAB-D: The next-generation simulation software," 2019. [Online]. Available: https://www.gridlabd.org/

[23] K. P. Schneider, Y. Chen, D. P. Chassin, R. G. Pratt, D. W. Engel, and S. E. Thompson, "Modern grid initiative distribution taxonomy final report," Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2008.