Privacy risk analysis of large-scale temporal data : application to electricity consumption data

Antonin Voyez, Antonin Voyez

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

Thèse de doctorat de

L'UNIVERSITÉ DE RENNES

École Doctorale N° 601
Mathématiques, télécommunications, informatique, signal, systèmes, électronique
Spécialité : *Informatique*

Par

Antonin VOYEZ

Privacy Risk Analysis of Large-scale Temporal Data : Application to Electricity Consumption Data

Thèse présentée et soutenue à Rennes, le 11/07/2023
Unité de recherche : IRISA (UMR CNRS 6074), Université de Rennes
Thèse N° :

Rapporteurs avant soutenance :

Engelbert MEPHU NGUIFO — Professeur, Université Clermont Auvergne
Nicolas TRAVERS — Enseignant-Chercheur, HDR - ESILV Engineering School

Composition du Jury :

Président :	Benjamin NGUYEN	Professeur, INSA Centre Val de Loire
Rapporteurs :	Engelbert MEPHU NGUIFO	Professeur, Université Clermont Auvergne
	Nicolas TRAVERS	Enseignant-Chercheur, HDR - ESILV Engineering School
Examinateurs :	Benjamin NGUYEN	Professeur, INSA Centre Val de Loire
	Marie-Christine ROUSSET	Professeure, Université Grenoble Alpes
Dir. de thèse :	Élisa FROMONT	Professeure, Université de Rennes
Co-dir. de thèse :	Gildas AVOINE	Professeur, INSA Rennes
Co-encadrant de thèse :	Tristan ALLARD	Maître de Conférences, Université de Rennes

Invité(s) :

Pierre CAUCHOIS — Ingénieur, Enedis
Olivier CHAOUY — Ingénieur, Enedis

Acknowledgement

Je tiens à remercier
I would like to thank. my parents..
J'adresse également toute ma reconnaissance à ....
....

Table of Contents

Résumé en français

Enedis est une entreprise de service public et le principal gestionnaire de réseau de distribution (GRD) français. Les GRDs sont responsables de la distribution de l'électricité en France. Les GRDs exploitent le réseau électrique basse et moyenne tension et gèrent les données associées. Dans le cadre de leurs missions, les GRDs collectent, analysent et publient les consommations individuelles d'électricité. Il s'agit notamment de permettre la facturation par les fournisseurs d'électricité des offres qu'ils proposent à leurs clients et de fournir aux clients des informations sur leur consommation d'électricité.

La France exploite le compteur intelligent *Linky* pour mesurer, collecter et publier la consommation d'électricité des ménages. Les GRDs ont l'obligation légale de partager les mesures au public. Certains tiers de confiance peuvent accéder aux mesures sans anonymisation toutefois le partage est principalement fait en utilisant des agrégats anonymisés. Cette politique restrictive est due à la sensibilité des données. Les mesures individuelles de la consommation d'électricité contiennent de nombreuses informations sur les habitudes d'un ménage. Pour donner un exemple, il est possible de détecter l'occupation d'une maison et les appareils électriques utilisés. Par conséquent, ces données sont considérées comme des informations personnelles en vertu des réglementations relatives à la protection de la vie privée (c'est-à-dire le RGPD). La violation de ces réglementations suite à une fuite d'informations personnelles expose le responsable de traitement (à savoir Enedis) à des amendes importantes et à des problèmes d'image. Cette thèse vise à comprendre les vulnérabilités en matière de protection de la vie privée liées à la publication de mesures anonymisées de la consommation d'électricité. En outre, cette thèse vise à fournir des pistes de réflexion pour la publication d'informations sûres et utiles sur la consommation d'électricité.

Le compteur intelligent est un élément essentiel de l'infrastructure de comptage. Il doit mesurer avec précision la consommation d'électricité à intervalles réguliers et la communiquer à son gestionnaire. La collecte automatique de données précises sur la consommation d'électricité est l'une des principales raisons justifiant le déploiement des compteurs intelligents. Pour les gestionnaires de réseaux (c'est-à-dire les opérateurs de distribution et de transport), les compteurs intelligents apportent une meilleure compréhension de la

consommation d'électricité. Pour les fournisseurs, ils permettent de développer des offres commerciales flexibles et une aide à la facturation. La transition énergétique est une autre raison justifiant l'utilisation des compteurs intelligents, car ils permettent aux individus de comprendre et d'adapter dynamiquement leur consommation. Les législateurs et les autorités locales utilisent les mesures de la consommation d'énergie dans la planification et l'exécution de nouvelles politiques publiques. En l'absence de compteurs intelligents, toute action sur le compteur (c'est-à-dire la mesure ou la modification du contrat) nécessite qu'un technicien se déplace physiquement jusqu'au compteur, ce qui limite les possibilités offertes par le compteur et, par conséquent, toutes les applications potentielles. La France, déploie le compteur intelligent *Linky* à la jonction entre le réseau électrique et le réseau domestique depuis les années 2010. Début 2023, environ 35 millions de compteurs, représentant près de 92 % des foyers français, auront été déployés.

Les GRDs sont légalement tenus de publier les mesures de la consommation d'électricité en utilisant des agrégats par seuil. Les agrégats sont un moyen peu coûteux et simple d'anonymiser des données. Dans cette thèse, nous nous concentrons sur les agrégats sous forme de *somme* et de *moyenne* dans la mesure où ils sont concernés par les publications actuelles. Il faut trouver un compromis entre la publication d'agrégats contenant des informations fines et la préservation de la vie privée. Contrairement aux données non agrégées, les agrégats ne sont généralement pas publiés seuls, mais avec des métadonnées: un ensemble de statistiques précalculées décrivant la population agrégée.

La quantité et la qualité des informations disponibles sur une série dépendent de sa précision. Les séries à la demi-heure contiennent plus d'informations que les séries quotidiennes (qui contiennent plus d'informations que les séries annuelles). La NILM (Non Intrusive Load Monitoring) est un domaine qui permet d'extraire des attributs des séries temporelles de consommation d'électricité. Nous notons que, dans leur état actuel, les techniques de NILM peuvent inférer des attributs sensibles tels que l'occupation d'un foyer et des paramètres socio-économiques sur des séries demi-horaires. Cependant, nous notons que les algorithmes de NILM nécessitent un entraînement qui requiert une vérité terrain importante (c'est-à-dire des séries étiquetées). La collecte de cette vérité terrain est complexe et coûteuse, ce qui limite (sans l'annuler) l'impact des attaques par inférence d'attributs basées sur la NILM.

Une autre menace pour la publication des séries temporelles de consommation d'électricité est le couplage de deux séries. Une série publiée au cours d'une période donnée peut-elle être liée à une autre série publiée au cours d'une autre période ? Nous pourrions imag-

iner qu'un attaquant dispose d'un accès détaillé à une série sur une petite période et d'une publication pseudonymisée de plusieurs séries sur une période plus longue. Trouver quelle série de l'ensemble de données pseudonymisées correspond à la série ciblée permet à l'attaquant d'acquérir des connaissances supplémentaires sur la cible.

Les menaces qui pèsent sur les agrégats sont différentes de celles qui pèsent sur les séries individuelles. Même s'il est possible de procéder à une inférence des attributs, le fait que les agrégats fusionnent plusieurs séries complexifie la tâche. Les caractéristiques individuelles se fondent dans la masse et finissent par être masquées par le nombre de séries agrégées. Les caractéristiques générales décrivant l'ensemble de la population remplacent les informations individuelles. La menace passe de l'extraction des caractéristiques à la déduction de la participation ou non d'un individu à l'agrégat. Les agrégats ne sont pas publiés seuls, mais sont accompagnés de métadonnées et d'informations descriptives. L'identification d'un individu comme faisant partie d'un agrégat lui attribue ces informations descriptives. Les attaques par inférence d'appartenance pourraient être généralisées à l'ensemble de la population de l'agrégat pour trouver toutes les séries à l'origine d'un agrégat.

Cette thèse se concentre sur les questions suivantes. Tout d'abord, nous étudions les modèles de menace pour la vie privée liés à la publication de séries temporelles agrégées de consommation d'électricité. Nous développons de nouvelles attaques de protection de la vie privée ciblant la publication d'agrégats. Ces attaques nous permettent de comprendre comment un acteur malveillant peut porter atteinte à la confidentialité d'un agrégat. En outre, les attaques permettent de connaître les exigences de l'attaquant. Tant en termes de connaissances de base que de puissance de calcul. Enfin, nous analysons les ensembles de données et les attaques, en essayant de comprendre ce qui rend un agrégat et une série individuelle vulnérables. Nous visons à fournir des connaissances pour choisir un seuil d'agrégation pertinent. En examinant les séries individuelles, nous visons à fournir une définition d'une série "atypique" qui devrait être retirée de l'agrégat. Cette thèse est organisée comme suit.

Tout d'abord, le chapitre 1 présente le contexte de la thèse : ce qu'est un compteur intelligent, les réglementations applicables et la sensibilité des données. Le chapitre 2 présente la littérature relative aux attaques contre la vie privée en se concentrant sur celles liées aux agrégats de séries temporelles de consommation d'électricité.

Dans le chapitre 3, nous présentons notre première contribution. Il présente les données de consommation d'électricité utilisées dans nos expériences. Ce chapitre présente

également une étude statistique de ces données. L'étude statistique contient une analyse de la vulnérabilité de la publication de séries individuelles (non agrégées) basée sur le taux d'unicité. Cette étude d'unicité montre que la grande majorité des consommations électriques sont uniques en considérant un très faible nombre de mesures. Les travaux de ce chapitre font l'objet de deux publications. L'étude statistique est publiée dans un rapport interne à Enedis. L'étude d'unicité est en cours de relecture par le journal *Nature Scientific Report, Smart Cities*.

Le chapitre 4 présente notre seconde contribution: l'attaque *SubSum*. Il s'agit d'une attaque par inférence d'appartenance permettant de trouver l'ensemble des séries à l'origine d'un agrégat. Cette attaque est basée sur une adaptation du problème de la somme d'un sous-ensemble. Elle requiert de nombreuses données (au moins celles présentes dans l'agrégat) ainsi qu'une importante puissance de calcul. Toutefois, cette attaque montre une forte capacité à identifier les séries présentes dans un agrégat. Nos expériences montrent que la capacité de l'attaque à identifier les membres d'un agrégat est lié au nombre d'individus connus, de la longeur de la série et du temps disponible pour l'attaquant. Si l'attaquant dispose de suffisamment de temps et de séries de longueur supérieur à la moitié de la taille de la population, il est généralement possible (dans plus de 90 % des cas) de retrouver les individus présent dans tous les agrégats formables à partir des séries connues de l'attaquant. Notons que le problème de la somme des sous-ensembles est un problème NP-difficile. De fait, il semble difficile d'appliquer cette attaque contre des populations importantes. Au maximum nous attaquons des populations de 4,500 séries nécessitant une journée de calcul par agrégat attaqué. Cette attaque est publiée dans l'édition 2022 de la conférence internationale *Conference of Security and Cryptography (SECRYPT)*.

Le chapitre 5 présente notre troisième contribution: l'attaque *STATS*. Il s'agit d'une attaque par inférence d'appartenance sur les agrégats de séries temporelles. Cette attaque permet de déterminer si une unique série est présente dans un agrégat. Cette attaque se présente sous la forme d'un problème de classification de séries temporelles. Elle entraîne un modèle de classification pour détecter la présence de la cible (appelé Shadow Training). Cette attaque requiert moins de connaissances préalables et de puissance de calcul que l'attaque *SubSum*. Elle montre de très bons résultats et est capable d'attaquer des agrégats de taille importante quand l'attaque est réalisée sur la même période que la période d'entrainement. Au maximum, nous attaquons des agrégats de taille 20,000. Lorsqu'elle est réalisée dans des conditions adverses (attaque sur une période différente que celle d'entrainement), l'attaque n'est plus en mesure d'attaquer des agrégats aussi importants.

Elle reste capable d'identifier jusqu'à 40 % des individus ciblés pour des agrégats de taille inférieure à 1,000. Cette attaque est rapide ne nécessitant que quelques minutes de calcul par agrégat attaqué. Dans ce chapitre, nous proposons aussi un score d'atypisme pouvant être utilisé pour anticiper la vulnérabilité d'une série à notre attaque.

Enfin, le chapitre 6 conclut la thèse en discutant les résultats et en proposant de futurs travaux intéressants. Ce chapitre propose aussi un résumé des travaux présentés lors de l'EnergyDataHack. Il s'agit d'un hackathon organisé par le Ministère des Armées visant à retrouver les datacenters français en utilisant (entre autres) les données de consommation électrique publiées par Enedis. Notre équipe a remporté la 1^{er} place de ce hackathon.

Finalement, nous répondons aux questions suivantes: Comment un acteur malveillant peut menacer la vie privée des membres d'un agrégat et qu'est-ce qui rend un agrégat vulnérable (compte tenu de sa taille et de sa population) ? Tout d'abord, cette thèse montre que toutes les attaques nécessitent l'accès à des séries individuelles. Par conséquent, la protection et le contrôle de l'accès aux séries individuelles sont essentiels pour prévenir les attaques sur les agrégats. La publication de séries individuelles pseudonymisées est risquée. Les séries individuelles sont hautement identifiables avec un minimum de connaissances préalables. Deuxièmement, alors que l'attaque *SubSum* nécessite beaucoup de temps et de puissance de calcul, l'attaque *STATS* peut être exécutée rapidement sur un ordinateur personnel. Cette thèse montre que le seuil d'agrégation actuel est insuffisant pour protéger tout le monde contre nos attaques. Pour protéger chaque individu contre l'attaquant le plus favorable (réalisant l'attaque au cours de la même période que l'entrainement), il faut significativement augmenter le seuil de publication. Nous ne formulons pas de recommandations pour un seuil capable de protéger tout le monde. Au maximum, nous attaquons des agrégats de 20,000 séries. À l'échelle de la population générale, cela pourrait encore représenter des milliers d'individus à risque. Cependant, seuls les individus atypiques restent vulnérables sur de grands agrégats. En outre, réaliser l'attaque dans des conditions moins favorable (c'est-à-dire attaquant une période différente que la période d'entrainement) ne permet plus d'attaquer de grands agrégats. Pour la plupart des séries ciblées, nous ne sommes pas en mesure d'attaquer des agrégats contenant plus de 1,000 séries. L'élimination des individus les plus identifiables des agrégats permettrait de réduire le seuil d'agrégation. Nous pensons qu'il est possible de réduire le seuil à 1,000 tout en protégeant plus de 90 % des séries contre l'attaquant le plus favorable (c'est-à-dire au cours de la même période). Notre score d'atypisme pourrait être utilisé comme mesure estimant la vulnérabilité de chaque individu présent dans un agrégat. L'élimination des

individus les plus identifiables des agrégats soulève toutefois la question de l'utilité des données sans introduire de nouvelles vulnérabilités. Il convient de noter que la population agrégée influence fortement les résultats de l'attaque. Il est nécessaire d'analyser chaque publication indépendamment pour détecter les vulnérabilités spécifiques aux membres de la population.

List of publications

Peer-reviewed international conference article:

Voyez, Antonin, Tristan Allard, Gildas Avoine, Pierre Cauchois, Élisa Fromont and Matthieu Simonin. "Membership Inference Attacks on Aggregated Time Series with Linear Programming." International Conference on Security and Cryptography (2022).

Papers currently under review:

Tristan Allard, Hira Asghar, Gildas Avoine, Christophe Bobineau, Pierre Cauchois, Elisa Fromont, Anna Monreale, Francesca Naretto, Roberto Pellungrini, Francesca Pratesi, Marie-Christine Rousset, Antonin Voyez. "Analyzing and explaining privacy risks on time series data: ongoing work and challenges" Communications of the ACM (2023, under review).

Voyez, Antonin, Tristan Allard, Gildas Avoine, Pierre Cauchois, Élisa Fromont and Matthieu Simonin. "Unique in the Smart Grid -The Privacy Cost of Fine-Grained Electrical Consumption Data" Nature Scientific Report Smart Cities (2022, under review).

Peer-reviewed national conference article:

Voyez, Antonin, Tristan Allard, Gildas Avoine, Pierre Cauchois, Élisa Fromont and Matthieu Simonin. "Attaque par inférence d'appartenance sur des séries temporelles agrégées en utilisant la programmation par contraintes" Conférence sur la Gestion de Données – Principes, Technologies et Applications (2021).

Source code:

`https://gitlab.com/phd_antonin/subsum` (alt. `https://gitlab.inria.fr/avoyez1/subsum`)

`https://gitlab.com/phd_antonin/tsunicity` (alt. `https://gitlab.inria.fr/avoyez1/tsunicity`)

`https://gitlab.com/phd_antonin/mia-ts` (alt. `https://gitlab.inria.fr/avoyez1/mia_stats`)

Chapter 1

Introduction

Enedis is a public service company and the leading French Distribution System Operator (DSO). DSOs are responsible for electricity distribution in France. DSOs operate the low and medium-voltage electricity network and manage the associated data. DSOs collect, analyze, and publish individual electricity consumption as part of their missions. The data is used by suppliers to bill their clients and to provide clients with insights into their electricity consumption. To the electricity network actors and collectivities, the analysis of precise electricity consumption information helps improve the management of the electricity network. Electricity consumption data plays a role in creating and applying energy policies.

France operates the *Linky* smart meter to measure, collect and publish households' electricity consumption. DSOs have the legal obligation to share the measurements with many recipients. Trusted third parties can access measurements, but only anonymized aggregates are usually shared. This restrictive policy is due to the sensitivity of the data. Individual electricity consumption measurements carry much information about the household's habits. To provide a non-exhaustive example, detecting whether a house's occupancy and the devices used is possible. Therefore, they are considered personal information under privacy protection regulations (i.e., the GDPR). Violating such regulations by leaking private personal information through the data sharing programs exposes the publisher (namely Enedis) to significant fines and bad public relations. This thesis aims to provide avenues of reflection for publishing safe and valuable electricity consumption information.

This chapter provides the context of the thesis. Section 1.1 starts by presenting the *Linky* smart meter, its history, and functions. Then, Section 1.2 looks at the relevant regulation as it governs the measurements collection and processing. Then, Section 1.3 presents the current anonymization method in place. We present in Section 1.4 the potential privacy threats over both un-anonymized and aggregated electricity consumption

time series. Finally, Section 1.5 presents the outline of this thesis: the research questions and our contributions.

1.1 Smart meters

The smart **meter** is a crucial component of the metering infrastructure. It must accurately measure the electricity consumption at regular timestamps and report it to its manager. The automatic collection of precise electricity consumption data is one of the main reasons justifying the deployment of smart meters. For the network managers (i.e., distribution and transport operators), smart meters bring a better understanding of electricity consumption. For the suppliers, they allow the development of flexible commercial offers and billing support. The energy transition is another reason justifying the usage of smart meters as they allow individuals to understand and dynamically adapt their consumption [1]. Legislators and local authorities use energy usage measurements in the planning and execution of new policies. Without smart meters, any action on the meter (i.e., measurement or changing the contract) requires a technician to move to the meter physically, thus limiting the possibilities offered by the meter and, subsequently, any potential applications.

Network actors

Client: An individual (household) or any entity (i.e., a company) consuming electricity and equipped with a meter.

Supplier: A company selling electricity to clients.

Distribution System Operator: The entity managing the electricity distribution to the clients.

Balance responsible parties: A private entity ensuring the balance between production and consumption within a limited perimeter (a supplier for example) [a].

a. https://www.services-rte.com/files/live/sites/services-rte/files/documentsLibrary/2022-09-01_RULES_MA-RE_SECTION_2_A-D_4200_en

1. https://observatoire.enedis.fr/

France, deploys the *Linky* smart meter at the junction between the electric grid and the household network. Lamb [Lam22] provides the history of the *Linky* program. Lamb traced the first works to digitalize meters back to the 80s. However, the first thoughts around the mass deployment of smart meters date from the early 2000s at the dawn of the energy market liberalization. We must wait until the early 2010s to see the first experimental deployment of the *Linky* system. The mass deployment started in 2015, intending to equip all households by 2020. More than 90 % (approx. 30 million) of households have been equipped by that date. In early 2023, around 35 million meters have been deployed. In this thesis, we focus only on the privacy questions [MRT12] related to publishing personal electricity consumption data.

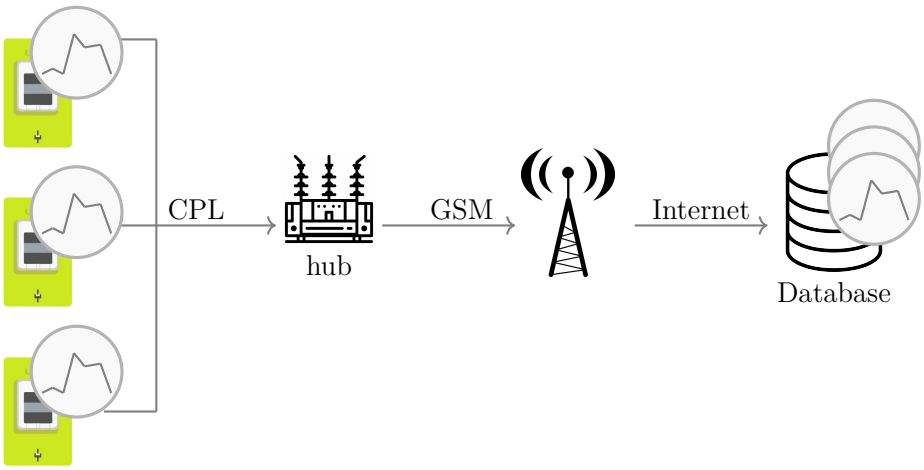


Figure 1.1 – Linky to database

The Linky meters work as follows (see Figure 1.1). The meter performs one measurement daily or every 30 minutes, depending on the client's choice. Daily measurements (also called "index") are measured in Watt-hour (Wh), and half-hourly measurements in Watt (W). Strictly speaking, the measurements, in Watt-hour , are called *consumption* and the half-hourly ones *power*. In this thesis, we abuse the language, call every measurement a **consumption**. Even if an individual can inject electricity onto the network (i.e., through solar panels), we consider that injected electricity is managed separately from the consumed electricity. No measurements can be negative. Plans are made to lower the measurement rate to 15 minutes. Since it has yet to be fully implemented, we only consider a minimum sampling rate of 30 minutes. The meter encrypts the measurements and sends them to a hub using the electricity distribution network with power line communication. The hub relays the information to the database using the internet. The database decrypts

the message and stores the measurements for further processing. Individual measurements form a **time series**: a sequence of timestamped measurements.

1.2 Regulation

Since the early 2000s, the European Union (under the directive relative to the electric market rules [09]) imposes the deployment of smart meters and the collection of energy consumption data. French laws (with article L322-8 of the "Code de l'Énergie" (Energy Code) [21] and article 29 of the Law "Transition Énergétique pour la Croissance Verte" (Energy Transition for Green Growth) [15]) implement the directive. Article 23 of the Law "pour une République Numérique" (for a Digital Republic) [16a] imposes the publication of energy consumption data to the public.

However, these laws also recognized the personal and sensitive character of the data and impose a treatment compatible with the current privacy laws: namely, the GDPR [16b] in Europe. In France, the Law "Informatique et Libertées" (Informatics and Liberties) [78] provides the legal baseline on privacy protection. Besides a right of information, rectification, and deletion, they state that any publication should be anonymized, such as it is impossible to re-identify any individuals included.

In France, articles D111-59 to D111-66 of the "Code de l'Énergie" (Energy Code) specify the anonymization method before publishing any electricity consumption series to the public. It requires the use of the **threshold aggregation** method. The **threshold** indicates the minimum number of series aggregated. The thresholds are summarized in Table 1.1. Publishing half-hourly measurements below 24 h (48 timestamps) requires aggregating at least 100 meters. Publishing periods between 1 and 31 days (1,488 timestamps) require aggregating at least 500 meters. Above, a publication must contain at least 5,000 series. For daily measurements, the publication must aggregate at least 100 individuals independently of the period published. The law also states that "atypical" measurements should be removed (without defining "atypical"). Publishing series containing only professional clients requires aggregating 3 series with a restriction that a single individual must not contribute to more than 85 % of the aggregate. The professional threshold is justified because they are not considered "personal information" contrary to the residential ones and, therefore, not subject to personal data protection laws. In this thesis, we focus on residential aggregates and series as they are directly concerned by the GDPR.

Frequency	Timestamps	Threshold
$> 24H$		≥ 100
$< 24H$	$< 24H$	≥ 100
$< 24H$	[24H; 31J]	≥ 500
$< 24H$	$> 31J$	≥ 5000
Professionals above 36 kW		≥ 3

Table 1.1 – Aggregate thresholds function of the publication according to the measurement frequency (daily or half-hourly) and the number of timestamp published.

1.3 Publishing aggregates

As the previous section shows, Enedis is legally bound to release electricity consumption measurements using threshold aggregates [17]. Aggregation is a traditional and still common method to publish data. Aggregates are cheap to compute (summing values) and require minimal hyperparameter tuning (the number of values to aggregate). In the rest of this thesis, we focus on *sum* and *mean* aggregates as Enedis' publications use them.

To give an example, Figure 1.2 shows three individual electricity consumption time series, the mean aggregate of the three series, and the mean aggregate of the whole *ISSDA* dataset (see Section 3.2). We see that the three series (*TS* 1 to 3 on Figure 1.2) have each a distinct consumption pattern. The aggregate reflects the characteristics of its population. Outstanding individuals have a substantial impact on small aggregates (*mean-aggregate* on Figure 1.2, performing the mean of TS 1 to 3). Individual patterns become indistinguishable by aggregating numerous series (*Global mean-aggregate* on Figure 1.2, aggregating 4622 series), replaced by global information about the general population. We see that the whole population consumes electricity during the day while *TS1* and *TS3* have two high consumption periods in the morning and the afternoon, strongly impacting the *mean-aggregate*. On the other hand, *TS2* has a tiny and stable consumption for most of the day; only its peak in the evening impacts the mean-aggregate.

There is a trade-off to find between publishing aggregates containing fine-grained information and preserving privacy. Unlike unaggregated datasets, aggregates are generally not published alone but with metadata: a set of pre-computed statistics describing the aggregate population (e.g., count, label). To make the published aggregates more valuable, they are usually associated with additional attributes characterizing the subpopulation in

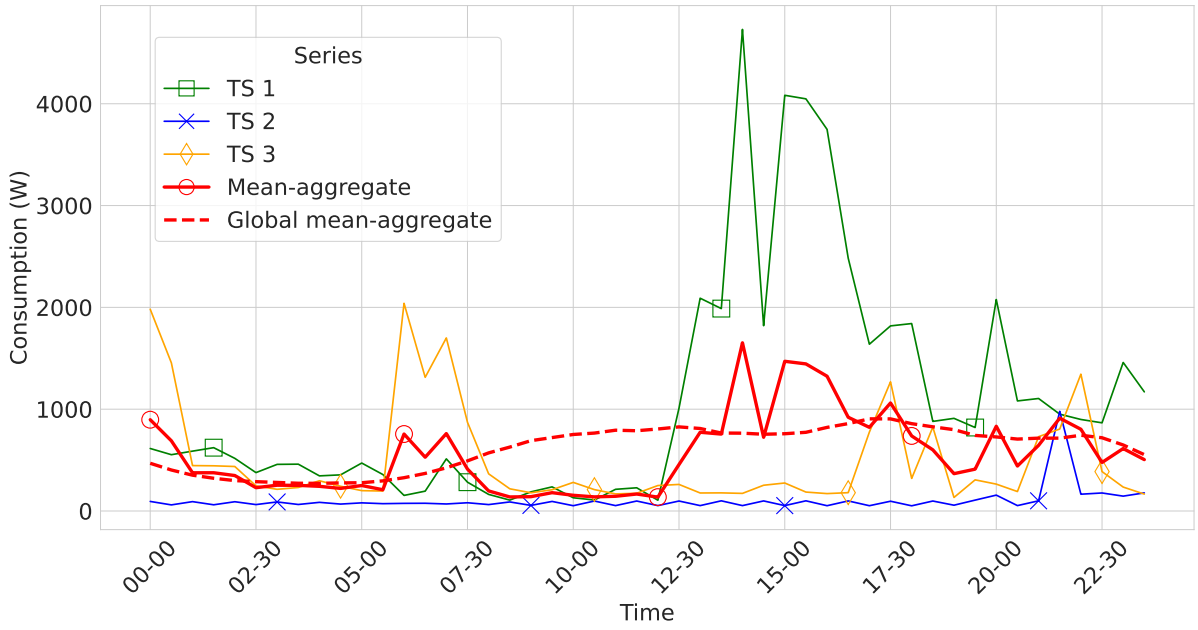


Figure 1.2 – Three illustrative half-hourly series over a single day, the mean-aggregate of the three series and the mean-aggregate of the whole population.

the aggregates. To illustrate this point, the current publications made by Enedis [2] consist of the average daily electricity consumption (time series) per region and profile (attribute). In the future, such characterizing attributes may potentially be, for example, the households occupancy [Che+13], socio-economical metrics about the household (i.e., household surface, number of inhabitants) [BSS13] or, as a generality, information extracted through NILM (see Section 2.1).

Despite the recognition of differential privacy [Dwo06; Dwo+06] as the de facto standard for privacy-preserving data publishing (see the box below), aggregation (without additional protection) is still widely used by the industry [ENA15; Pod21] and statistical entities [3] [4] to publish protected data. Aggregates have some advantages over differential-privacy-based methods. One of them is that aggregates allow the publication of unbounded time series. Publishing timestamped data while respecting differential privacy requires distributing the privacy budget (ϵ) according to the number of timestamps. With a large

2. `https://data.enedis.fr/`
3. `https://www.insee.fr/fr/information/4174982`
4. `https://www.census.gov/library/visualizations/2019/comm/history-privacy-protection.html`

number of timestamps, the expected variance of the noise increases significantly and reduces the publication utility.

However, despite being widely used as a publication method, it is still challenging to justify the threshold choice [WD96; ENA15; ISO18]. Few recent works propose to attack aggregates (see Section 2). One of the objectives of this thesis is to test the current threshold defined by the regulation to find potential vulnerabilities or provide detailed insight justifying the choice of pertinent threshold and defining "atypical" individuals.

ϵ-differential privacy

Formally, introduced by Dwork is 2006 [Dwo06; Dwo+06], ϵ-differential privacy is the de-facto standard for the publication of privacy sensitive information. Given a function f, an observer should not be able to tell if an individual participated in the result of f. It is done by adding noise, simulating the potential impact of a single individual on the result (with a margin of error). In addition, the privacy budget (ϵ) calibrates the magnitude of the noise. If the noise is too weak, the publication might be vulnerable. If the noise is too strong, it destroys information, and the publication loses its utility.

1.4 Publication sensitivity

As the previous sections show, electricity consumption time series are sensitive and require adequate protection. This section details the privacy threats faced by both individual series and aggregates. This section details the actors having access to individual series and the ones having access to aggregates. Finally, we provide accurate threat models to credibilize the attacks presented in this thesis.

1.4.1 Threat on individual series

In this section, we discuss the electricity time series sensitivity. We do it for two reasons: First, we study in Section 3.5 the vulnerability of publishing time series against re-identification attacks. Second, every attacks against aggregates need access to individual

time series. Therefore, the question of obtaining access to individual time series is a prerequisite to the realization of the attacks.

We identified the actors having access to individual electricity consumption time series. They could either behave maliciously or leak the data to a malicious actor. Even if all actors are honest and adequately protect the entrusted data, data leaks are a threat to be taken seriously. Leaks can be triggered by an external actor or through a malicious employee. Data leaks multiplied in recent years with more than 5,000 breaches referenced in 2021 [5] (+ 44 % in 2022 [6]). Such numbers are the tip of the iceberg, and many leaks remain unaccounted for. Without leaks, the attacker could collect series directly at the source (from the home electricity network) or through the third parties accessing the measurements.

The quantity and quality of information available on a series depend on its precision. Half-hourly series carries more information than the daily ones (which carry more information than yearly ones). NILM (Non-Intrusive Load Monitoring) is a data science field that infers attributes from electricity consumption time series. We detail NILM in Section 2.1. Still, we note that in its current state, NILM techniques can infer sensitive attributes such as home presence and socio-economic metrics on half-hourly series. However, we note that NILM algorithms require extensive ground truth information (i.e., labeled series) for training. Gathering such ground truth is complex and expensive, limiting (without canceling) the impact of NILM-based attribute inference.

Another threat to the publication of electricity consumption time series is the linkage of two series together. Given a series published during a period, can that series be linked to another one (from the same individual) published in another period? We could imagine an attacker having detailed access to a series (and subsequently the whereabouts) over a small period and a pseudonymized publication (by removing Identifying information) of multiple series over a longer period. Finding which series in the pseudonymized dataset corresponds to the targeted series allows the attacker to gain additional knowledge of the target.

5. `https://www.verizon.com/business/resources/reports/dbir/2021/data-breach-statistics-by-industry/`

6. `https://www.proofpoint.com/us/resources/threat-reports/cost-of-insider-threats`

1.4.2 Threat on the aggregates

Every aggregates are considered public as they are published respecting the threshold. The threats to the aggregates are different from those to the individual series. Even if it might be possible to perform an attribute inference attack, the fact that aggregates merge multiple series complexifies the task. Individual patterns will blend in and ultimately be masked with the number of series aggregated. General characteristics describing the whole population replace individual information.

The threat shifts from extracting characteristics to inferring whether or not an individual participates in the aggregate. Aggregates are not published alone but are accompanied by metadata and descriptive information. Identifying an individual as part of an aggregate labels it with that descriptive information. Membership inference attacks could be generalized to the whole aggregate population to find all series at the origin of an aggregate.

1.4.3 Attackers motives

We identify three main families of objectives that could motivate a malicious actor to attack electricity consumption published data.

Public relation issues: Performing an attack to hurt the company's image is one of the main threats faced at the moment. Malicious actors might also try to blackmail the company for monetary gains. Either they get paid a ransom, or they publish the sensitive data at their disposal. Even if no such attacks have been performed against anonymized datasets, they are already widely performed through data leaks [Phi17] [7].

Buisness gains: The current electricity market is very competitive, with many suppliers fighting each other to gain market shares. Therefore, using borderline (and sometimes illegal) methods to gain or retain a client is tempting. Multiple cases of mismanagement of private data [8] and aggressive marketing [9] from suppliers have been documented. As we previously identified, suppliers could have access to detailed measurements of their (former) clients. We note in particular that if it is

7. https://www.cloudflare.com/learning/insights-ransomware-extortion/
8. https://www.cnil.fr/fr/prospection-commerciale-et-droits-des-personnes-sanction-de-1-million-deuros-lencontre-de
9. https://www.economie.gouv.fr/dgccrf/sanction-lencontre-de-la-societe-eni-gas-power-france-pour-non-respect-du-droit-de

possible to identify a former client in a new publication, it is possible to understand the reason for his departure and make a tailored new offer to regain it.

Suppliers are not the only ones with financial interests in attacking published electricity consumption time series. Gaining in-depth knowledge of a household consumption pattern could prove extremely valuable for advertisers.

Spycraft: Individual and commercial consumption measurements carry much personal information ranging from the home presence, devices used, and company activity. Therefore, they make a tempting target for various spies. Against companies, finding the location of factories, the machinery they used, and how they are used could prove helpful for economical spies [10].

1.5 Outline

This thesis is funded by the Enedis company to study the privacy challenges posed by the publication of (aggregated) electricity consumption time series. In this introduction, we discuss the privacy threat models related to aggregated electricity consumption time series publication. *Why* would someone conduct a malicious action to extract sensitive information from electricity consumption time series? This thesis focuses on the following questions.

1. We develop privacy attacks targeting the publication of aggregates. With these attacks, we aim to understand *how* a malicious actor can breach the privacy of an aggregate. Besides, the attacks provide knowledge of the attacker's requirements. Both in terms of background knowledge and computational power.
2. We analyze the datasets and the attacks, trying to understand what makes an aggregate and an individual series vulnerable. We aim to provide knowledge for choosing a relevant aggregation threshold. We aim to define an "atypical" series that should be removed from the aggregate.

This thesis is organized as follows. First, Chapter 1 presents the context of the thesis: the smart meter, the related regulations, and the data sensitivity. Chapter 2 presents the literature related to privacy attacks. The chapter focuses on the most relevant attacks related to aggregates. Chapter 3, presents our first contribution. We present the electricity

10. `https://www.capital.fr/economie-politique/espionnage-industriel-les-affaires-qui-ont-fait-trembler-l-economie-1074640`

consumption datasets used in our experiments. Besides, it presents an in-depth statistical study of these datasets. This statistical study presents a vulnerability analysis of individual (un-aggregated) electricity consumption time series publications. Chapter 4 presents our second contribution, the *SubSum* attack. It is a membership attack aiming to find all the series participating in an aggregate. Chapter 5 presents our third contribution, the *STATS* attack. This attack finds if a series participates in an aggregate. Finally, Chapter 6 concludes the thesis by discussing the findings and providing interesting future works.

Chapter 2

Literature review

In the last decades, numerous privacy attacks trying to infer sensitive information from published data have been proposed. This chapter survey privacy attacks developed and highlights the most relevant ones in the context of this thesis. Based on the classification made by Dwork, Smith, Steinke, and Ullman in their survey [Dwo+17], we can distinguish 4 main attack families with different goals, background knowledge, and methods:

1. **Attribute inference:** The attacker aims to extract privacy-sensitive attributes from the published data (Section 2.1).
2. **re-identification:** The attacker aims to link an anonymized (or pseudonymized) record back to its original owner (Section 2.2).
3. **Reconstruction:** The attacker aims to reconstruct the dataset at the origin of an anonymized publication (Section 2.3).
4. **Membership Inference:** The attacker aims to detect whether or not a target individual was used in the construction of a published dataset (Section 2.4).

2.1 Attribute inference attacks

Attribute inference attacks aim to extract privacy-sensitive attribute from published data. The **attribute** is an information labeling the data. Attributes inference attacks are not always considered as a privacy violation but rather as legitimate data science [1]. However, it is possible to infer privacy-sensitive attributes directly from the dataset by crossing the information contained in multiple datasets [GL16].

In this chapter, we focus on **N**on-**I**ntrusive **L**oad **M**onitoring (NILM) aiming at extracting valuable information from electricity consumption time series [Fau+17]. Hart presented the first NILM paper in 1992 [Har92]. It is also referred to in the literature as Non-Intrusive Appliance Monitoring.

1. https://differentialprivacy.org/inference-is-not-a-privacy-violation/

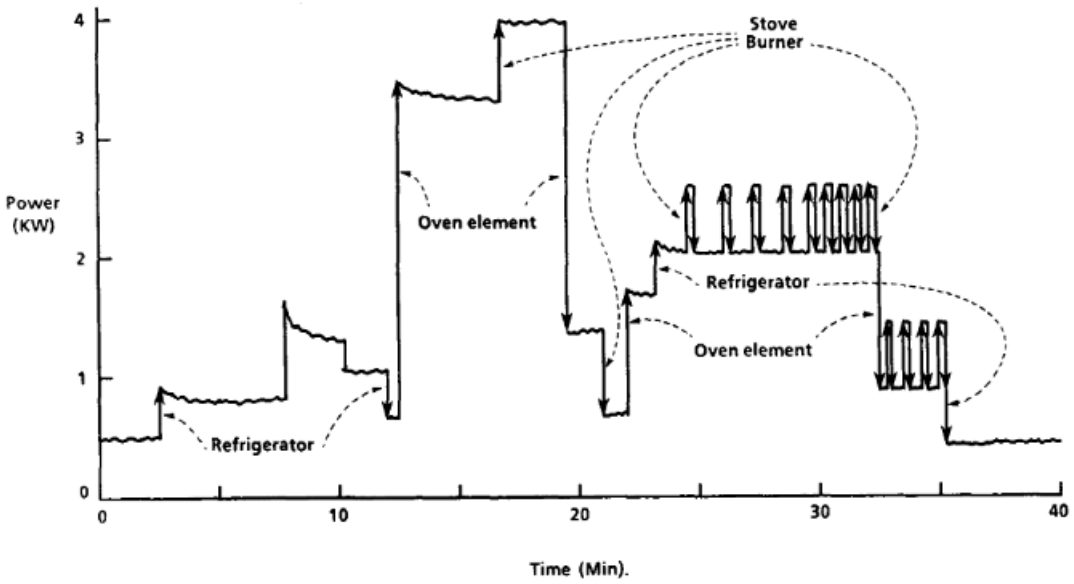


Figure 2.1 – NILM example at infering devices used [Har92].

Faustine et al. [Fau+17] surveys NILM techniques. For example, using NILM, it is possible to detect which electrical devices are used and when by a given household [Har92; Mol+10; LMW10; AR13; VKD13; ACF12; SM21]. Figure 2.1 illustrates the concept of NILM. The figure shows an electricity consumption time series labeled with the devices using electricity at a given time. With NILM, it is possible to create consumer profiles and improve contracts and grid management [Chi+03; Bir+12]. Some works predict the thermal sensibility of a building [Bir+12], a metric often used in thermal renovation and for the energy transition. [GJL12] detects the TV program currently watched.

However, some extracted information can threaten individuals. For example, detecting whether a house is occupied [Che+13; AR13; JJS17] could help burglars find empty homes. Another example would be to extract sensitive socio-economic metrics on households [LMW10; AR13; BSS13; MH20; SM21] such as the number of residents, the family status (single, number of children), the employment status (employed, unemployed or retired), the social status and even the sleep patterns. Therefore NILM techniques can be diverted by a malicious actor to extract sensitive attributes from published electricity consumption time series.

Note that several drawbacks make NILM challenging to be used in an attack. Firstly, many NILM techniques inferring detailed insights on the household whereabouts are performed on time series with a sampling rate below a minute for [Mol+10; VKD13; GJL12;

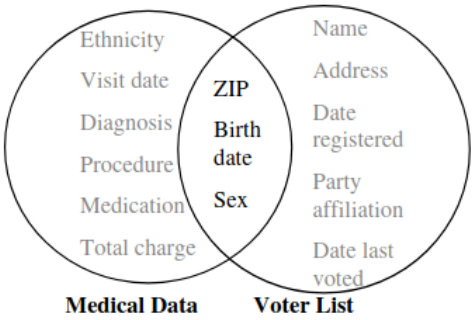


Figure 2.2 – re-identification attack by crossing unique attributes between two datasets [Swe00].

Bir+12; Rot+19]. It is incompatible with the current precision of the smart meters (performing one measurement every 30 minutes). Secondly, all NILM methods require training and ground truth information. The **ground truth** corresponds to a series correctly labeled with the information to extract. Obtaining the ground truth is particularly challenging and expensive. A simple survey linking a series to an attribute could be sufficient. For example, Beckel [BSS13] uses the survey accompanying the ISSDA dataset to infer the survey answers from the series. However, most of the time, gathering the ground truth requires obtaining the authorization of the individuals and installing intrusive physical devices. Most works previously mentioned trains their algorithms on a limited number of households [VKD13; ACF12; Bir+12; Rot+19]. Only a few works use the (relatively) numerous and coarser-grained *ISSDA* dataset [BSS13; MH20].

Because of these drawbacks, we decided not to pursue a general research direction about NILM and attribute inference attacks. We do not rule out an attacker getting access to sufficient knowledge by collecting the ground truth or through pre-trained third-party models to perform a NILM-based attack.

2.2 Re-identification attacks

Re-identification attacks link information in a pseudonymized dataset to a real individual's identity. **Pseudonymization** (or de-identification) is a technique often used to anonymize a dataset by removing (or replaced by a pseudonym) any personal information from a dataset. With pseudonymization, performing the same analysis as on the raw dataset is possible.

The most impressive re-identification attacks happen when individuals are unique. An individual is said **unique** if it is the only one in the dataset with a set of descriptive variables (called **quasi-identifiers**). The **uniqueness** is defined as the proportion of unique individuals (relative to a set of quasi-identifier) in the dataset.

Re-identification attacks suppose the attacker has access to a pseudonymized dataset containing sensitive information and a re-identification dataset containing identity information. An attacker can link sensitive information to an identity by crossing unique quasi-identifiers between the two databases. The re-identification attack could be summarized (and is often implemented) as an SQL join between two tables on a shared set of columns.

In 2000, Sweeney [Swe00; Gol06] showed that 87 % (216 million out of 248 million) individuals were unique in a pseudonymized medical database only by knowing three characteristics: the ZIP code, gender, and birth date. Sweeney used a public register containing the information of 54,000 individuals to find the names behind a pseudonymized medical dataset using the unique characteristics shared between the two databases (see Figure 2.2). Using this method, Sweeney re-identified W. Weld, the Massachusetts governor at the time [Swe02]. Similarly, Narayanan and Shmatikov [NS08] crossed the public IMDB database containing user film ratings with the pseudonymized Netflix one, re-identifying users behind Netflix accounts.

Several studies pointed out high uniqueness over sparse datasets without proposing a full attack. Uniqueness is used as a risk measurement metric. High uniqueness with minimum identifiers facilitates the re-identification (as shown by Sweeney, Narayanan and Shmatikov). De Montjoye et al. performs such uniqueness study on a countrywide mobility dataset containing 1.5 million traces [De +13; DP16] over 15 months. A **trace** is the timestamped position of an individual. In this dataset, a position corresponds to a GSM antenna. Therefore, each antenna encodes multiple individuals' locations during the same period. There are, on average, 2000 inhabitants per area covered by an antenna. The authors found that four randomly chosen points are enough to characterize 95% of the individuals in the dataset uniquely. Two years later, De Montjoye et al. reiterates a new uniqueness study on payment data [Mon+15]. This time, the database contains the credit card history of 1.1 million individuals for three months. The authors found that randomly taking four transactions per individual was enough to uniquely identify 90 % of them uniquely. In both cases, the authors found that reducing the dataset precision does not prevent a high uniqueness. They reduced the precision by aggregating multiple

positions (or shops in the second attack) and timestamps. With the transaction dataset, 10 points are enough to identify 80 % of the population when aggregating the transactions over two weeks and 350 shops.

Without uniqueness, re-identification attacks are performed on GPS traces [GH05; GKP13]. The goal is to link pseudonymized GPS traces to identified ones using a likelihood ratio. In the first case, [GH05] compute a likelihood ratio between traces. In the second, [GKP13] model the traces as a Mobility Markov Chains. Then the authors measure the distances between the chains and link the closest together to perform the re-identification.

Jawurek et al. [JJR11] propose two re-identification attacks tailored against pseudonymized electricity consumption time series. Both attacks aim to link a pseudonymized time series with a non-pseudonymized one. The first attack finds anomalies in both datasets simultaneously (requiring that both datasets overlap) to perform the linkage. Anomalies are detected by computing a distance to an individual nominal consumption (mean for the period). The second attack uses pattern recognition to find common patterns that could identify an individual (thus not requiring overlap). The recognition is done by training an SVM classifier for every individual to re-identify.

2.3 Reconstruction attacks

Reconstruction attacks try to restore all or parts of the data at the origin of an anonymized publication. Dinur and Nissim introduced them in 2003 [DN03]. The anonymized data could be published once (i.e., as a CSV file) [GAM19; MSS20] or through a set of queries (i.e., an SQL interface) [FEM17; CN20]. All reconstruction attacks in the literature [DN03; DMT07; KRS13] share the same concept: the attacker uses the available data to build a set of linear constraints modeling the data to reconstruct. A **solution** is a set of individuals (with their values) matching the constraints. If the modeling is sufficient, then the constraints produce a single solution corresponding to the original data. In the worst case, multiple solutions match the constraints. When the set of solutions is reasonable, it offers valuable insight into the original data.

Garfinkel et al. [GAM19] propose a reconstruction attack on the US census statistics published per geographical area. The attacker dataset consists of a small population (i.e., a US county) where the same characteristics (here: sex, race, and marital status) are described by multiple metrics (here: the number of people, the mean, and the median

age). Having the same characteristics described by multiple metrics allows the attacker to build numerous constraints covering all characteristics and metrics. Besides, the attacker can build multiple constraints per individual. Only a few combinations (i.e., characteristics per individual) match the constraints, leading to a small set of solutions potentially at the origin of the publication.

Of course, obtaining a set (greater than one) of potential solutions prevents the reconstruction of the original dataset. The attacker could compute a probability of presence in the original dataset per individual. The attacker obtains knowledge about the original population if the probabilities are better than random guessing. Besides, the attack is limited by the number of constraints possible given the set of characteristics and metrics available. However, the following paragraph shows that the only limitation is the number of constraints the attacker obtains. Adding more constraints reduces the resulting set of solutions. Ultimately, only one solution matches the constraints: the one at the origin of the publication.

In [CN20], Cohen and Nissim propose an attack on the *DIFFIX* [FEM17] system. *DIFFIX* is a database query system where users perform custom *SQL* counting queries over a private dataset. *DIFFIX* maintain privacy by limiting the minimal number of rows queried, adding noise, and restricting the usage of some *SQL* operands. The attacker constructs a large set of linear constraints by submitting a large batch of queries using well-chosen pseudo-random filters, fully restoring the original database. In this attack, the filters are designed pseudo-randomly to cover the whole database population with multiple constraints. Besides, the attacker predicts which individuals are present in the filter. Individuals are represented with a unique database id (i.e., an integer). No information about an individual true existence (the id might be contained within the filter but not present in the database) is available to the attacker. Solving the constraints confirms the individual existence and provides its value.

Martinez, Sébé, and Sorge [MSS20] propose a reconstruction attack on electricity consumption data. The attacker access half-hourly pseudonymized consumption measurements and monthly aggregates containing customer information. New pseudonyms are generated for every timestamp, meaning tracing an individual consumption series from the measurements is impossible. The aggregates are computed by each meter by summing the measurement of multiple timestamps. The attack is modeled as a knapsack problem. It aims at building credible series (linking individual measurements over time) using the set of individual measurements and aggregates. For every aggregate, the attacker builds a set

of knapsack constraints, such as the sum of the individual consumption constituting the aggregate. This paper is adapted by Dietrich et al. [DLS20] on the UMass [2] electricity consumption dataset containing only 114 series. Both attack implementations focus on a tiny period: below 60 timestamps for [MSS20] and below 15 for [DLS20]. Their implementations are computationally intensive, and attacking more significant problems exceeds their available computational power (24 cores and 512 GiB of RAM). Besides [MSS20], [DLS20] studied the impact of the measurement precision on the attack success and studied the number of solutions found. They show that more precise measurements are more vulnerable to reconstruction attacks.

Reconstruction attacks can only occur in a setup with extensive background knowledge available to the attacker. For example, in the *DIFFIX* attack [CN20], 3500 queries were required to reconstruct a subset of 73 individuals. Also, solving the linear programs modeling the constraints is an NP-hard problem. However, linear reconstruction attacks are powerful when applicable as they can restore the whole private dataset or give valuable insight into a potential set of individuals forming the dataset.

2.4 Membership inference attacks

Membership **I**nference **A**ttacks (**MIA** in short, also referred by [Dwo+17] as *tracing attacks*), regroup a wide range of attacks aiming to find whether an individual is present in a publication. MIA often targets machine learning models or aggregates. MIA was introduced by Homer et al. in 2008 [Hom+08]. On this attack, the goal was to find if a gene was present in a set of gene subsequences. We split this section into two categories, one about MIA against machine learning models and one presenting the works tailored against aggregates.

2.4.1 MIA on machine learning model

Most recent works around MIAs focus on the following problem: given a machine learning model and a targeted individual: find if this individual is used to train the machine learning model. A recent survey [Hu+22] reports more than a hundred articles discussing MIAs against machine learning models. In this section, we present a few influential of these and focus on the works and techniques potentially usable against aggregates.

2. `https://traces.cs.umass.edu/index.php/Smart/Smart`

In 2017, Shokri et al. [Sho+17] introduced the influential **shadow model** concept. This concept is largely used and extended [Tru+21; Che+22]. In this attack, the attacker has access to an individual (the target) and black-box access to a model. A black-box access means the attacker can only ask the model for predictions without accessing any information about the model (i.e., algorithm, training data). Besides, the paper considers the attacker has access to similar data as the one used to train the model. The idea of the shadow model is to use the target and similar data to train multiple fake models (both trained with and without the target). Then, an attack model is trained to detect which model has been trained using the target. The attack model is then used to predict the target's presence in the real model.

The [Sho+17] attack uses the accuracy, the precision, and the recall as success metrics (defined in Section 5.1). The **accuracy** is the ratio of correct classification (both positive and negative) during the testing phase. The **precision** is the proportion of correct positive classifications. The **recall** is the proportion of positive labels correctly identified by the model. The **F-score** (the harmonic mean of the precision and the recall) could also be used as a success metric [Rah+18]. Compared to the accuracy which shows the probability of getting a correct classification, the F-score offers a better vision of the True Positive (i.e., the examples correctly classified as part of the training data) which can be useful for the attacker. One attacker might consider that he must not accuse an innocent (i.e., having a False Positive). In that case, the attacker performs the attack only when the precision is high.

Salem et al. [Sal+19] demonstrate the attack feasibility by training a single fake model to simulate the targeted model. This model is trained to imitate the attacked model using a similar dataset to the one used to train the target model. Here the model is trained on half of the similar dataset. The model then classifies the remaining individuals in the similar dataset and the attacker observes the prediction probabilities. Based on the prediction results, the attacker builds a new model to detect if targeted records have been used to train the targeted model.

Based on [Sho+17], Yeom et al. [Yeo+18] explore the reason behind MIA's success against machine learning models. The authors found that model overfitting is one of the main factors influencing MIA success. A model is said to **overfit** when it remembers the training values instead of learning a general trend. Another factor is the importance taken by some individuals in the attacked model. The most important individuals are the most vulnerable. Overfitting is not much of a concern in dealing with aggregates as

it is a machine-learning concept. An aggregate (sum or mean) learns nothing about the data. The weight of each individual in the final publication is particularly relevant while dealing with aggregates. As seen by Figure 1.2 some individuals have more impact on the aggregate than others. [Yeo+18] also introduced the **membership advantage** (Adv) metric defined in Formulae 2.1 as the attack advantage over random guessing.

$$Adv = 2 \times accuracy - 1 \tag{2.1}$$

Jayaraman et al. [Jay+21] propose an improvement of the [Sho+17] and [Yeo+18] attacks to support imbalanced datasets for training the shadow model. Besides, this paper explores MIA against differentially private models evaluating the impact of the privacy budget (ϵ) on the membership inference. Unsurprisingly, a lower budget reduces the ability to perform MIA. However, a lower budget significantly reduces the model's utility. There is a tradeoff to find between the model resistance to MIA and its utility.

While the previous works quantify the attack success through accuracy-based metrics, Long et al. [Lon+20] propose a pragmatic approach. Their idea is to force the attack model to reach high *precision*. The authors consider the attacker must not accuse innocent (i.e., perform a classification when there is a chance of getting a False Positive). The attacker makes a prediction only when the attacking model is confident enough. The attacker does not predict anything when the model confidence is too low. The model confidence is computed using p-values. A model is deemed confident enough when the p-value is below 0.05. This attack shows that even if an MIA does not reach a high accuracy, it still is possible to identify the presence of a few individuals with good precision and high confidence.

GANs [3] are increasingly used to circumvent privacy issues by generating credible fake data. As they produce synthetic data similar to personal ones, it is believed that the synthetic data could be safely published without restriction. MIA has been developed to study if real private information, used in the training phase, could leak from the generative model and the resulting synthetic data [Hay+17; HAP17; Mel+19; SOT22]. These works found that GANs have the same privacy weaknesses as the other machine learning models.

3. **G**enerative **A**dversarial **N**etworks (GAN): A class of machine learning putting in competition two networks. One generates data and the other detects if the data is fake or real. The goal is to generate a model producing fake data indistinguishable from the real ones.

2.4.2 MIA on aggregates

In 2015, a report from the Energy Network association [4] claimed that aggregating two electricity consumption time series was enough to guarantee the privacy of a publication. According to them, it is impossible to distinguish an individual's contribution to an aggregate. Buescher et al. [Büs+17] refuted this claim and propose a new membership inference attack. Contrary to the previous MIA, this attack is modeled as a cryptographic game. In that game, the attacker is given a pair of series and must find which one is present inside the aggregate. The attack uses simple statistical estimators (Pearson correlation, Mean Squared Error, and the most common consumption peaks). It is evaluated using the membership advantage metric. This attack achieves almost absolute success (advantage above 75 %) against an aggregate of size 2. The paper explores the attack's success against aggregates of sizes up to 50. The advantage becomes low (below 25 %) for aggregate sizes above 20.

In his PhD thesis [Pyr19; PTC17; PTC18; PTC20] and the related papers, Pyrgelis and his co-authors conducted an in-depth MIA study on aggregated location data. His attack is based on the shadow model principle (described above in [Sho+17]). Individual mobility traces are represented as a boolean sparse matrix (the cell is set to 1 if the individual was present at that location and at that time). The aggregate consists of the sum of multiple individual matrixes indicating the number of individuals per location and timestamp. The attack consists of simple machine learning models (Logistic Regression, Random Forest, K-Nearest Neighbors, and Multi Layers Perceptrons) taking simple features (min, max, mean, median, variance, and standard deviation) computed for each location over the targeted aggregate. In its latest variant [PTC20], the attack also uses PCA [5] to compute features and uses only Logistic Regression deemed the most efficient machine learning method. Therefore, this attack does not consider the data's temporal aspect. The attack challenges aggregate sizes up to 100. The attack is very successful for small aggregates (5, 10). The success rate goes down when the aggregate size increase. The authors also study many potential protection methods and utility metrics. They find that differential privacy methods achieve good privacy but at the cost of the data utility.

4. https://www.energynetworks.org/industry-hub/resource-library/smart-meters-data-aggregation-assessment-final-report.pdf

5. Principal Component Analysis

Chapter 3

Datasets analysis

This chapter presents and analyses the datasets used in this thesis: the *ISSDA* [CER12] dataset, the *LONDON* [Net13] dataset and the *ENEDIS* dataset. *ISSDA* and *LONDON* are two public electricity consumption datasets. We use them as a reproducible caution for our work. However, they are relatively small, around 5000 series each, preventing us from running large-scale experiments on them. We use the *ENEDIS* dataset containing more than 3 million half-hourly series and more than 30 million daily series.

This chapter offers a statistical study of each dataset. Our primary objective is to explore the datasets, understand the consumptions and find relevant patterns. We observe the consumption patterns from the yearly level down to the half-hourly level. We study the factors influencing consumption. This study highlights the importance of temperature as a main factor influencing consumption. We find higher consumption during the winter than in summer. We compare the average consumption to the average temperature and find a strong correlation between consumption and low temperature. We observe a consumption peak during the evening after working hours. Finally, we observe that higher average consumption leads to a higher dispersion. Although the *ISSDA* and *ENEDIS* datasets behave similarly, the *LONDON* dataset is different, with higher consumption during the summer and the night.

Finally, in this chapter, we study the vulnerability of the datasets to a re-identification attack by looking at the uniqueness metric. **Uniqueness** is a widely used measure for evaluating the vulnerability to re-identifications of personal data. For example, the famous re-identification of Governor Weld performed by Sweeney within a health dataset [Swe02] is possible because Governor Weld's record (`date of birth`, `zipcode`, `sex`) is unique in the health dataset disclosed. So any adversary knowing the (`date of birth`, `zipcode`, `sex`) triple of Governor Weld can join it with the dataset to obtain his health information. Obviously, the rate of unique measurements impacts the re-identification. Uniqueness is used in the Netflix attack [NS08]. It is exploited by large-scale empirical studies on mobility traces [De +13] and transaction data [Mon+15]. Uniqueness is now well recognized as a

risk measurement metric [Sek+21; RLK21]. Based on previous uniqueness works [Swe02; NS08], we measure uniqueness as a re-identification risk metric. In this context, uniqueness quantifies the fraction of households that can potentially be re-identified by accessing part of their electricity consumption time series.

To the best of our knowledge, this work is the first uniqueness study performed nationwide on fine-grained electric consumption time series. Previous large-scale uniqueness studies [NS08; De +13; Mon+15] do not mention electric consumption time series. Related works [JJR11; Buc+13; DLS20; Büs+17] focusing on the vulnerabilities to re-identification of electric consumption time series do not perform any in-depth uniqueness study. A recent work [Cre+22] focuses on re-identifying individuals when the adversarial background knowledge is different from the period of the disclosed dataset. However, uniqueness is not studied in [Cre+22], and the dataset consists of interactions between individuals (e.g., who calls who).

In a nutshell, we observe that even at such a large-scale, most electric consumption time series are unique when considering a few consecutive electric consumption measures. For example, on average, 90 % of the time series are unique when considering only 5 consecutive daily measures. Moreover, tremendously degrading the precision of the electric consumption measures is not enough to hinder possible re-identifications, even when rounding the measures from the watt to the kilowatt on half-hourly consumption measures.

The *ENEDIS* datasets statistical study (Section 3.4) is published as an Enedis' internal report. The uniqueness study is an independent article currently under review by the *Nature Scientific Report Smart Cities* journal [Voy+22b].

We start this chapter in Section 3.1 by exposing data management regulations and the analysis methodology. Then, we present the statistical study of the three datasets, starting with the *ISSDA* dataset (Section 3.2), followed by the *LONDON* dataset (Section 3.3) and the *ENEDIS* dataset (Section 3.4). Finally, we present the results of the uniqueness study in Section 3.5. Section 3.6 discusses the results and concludes.

3.1 Practical information

3.1.1 Legal compliance

This thesis leverages collecting and manipulating a significant amount of electricity consumption time series. They are considered personal information and have been managed under the applicable legislation [78; 16b]. In this document, we only share individual series from public sources.

The two public datasets (*CER-ISSDA* [CER12] and *London* [Net13]) are managed regarding their respective license. Both contain pseudonymized data and are collected with the agreement of the concerned individuals. All requests should be addressed to the provider.

We also use data from our industrial partner: Enedis. They are collected and managed as part of the normal process of the company [1]. Daily consumptions (called *index*) are systematically collected and transmitted to the electricity provider for billing purposes. They are kept for 72 months before being deleted. The collection of half-hourly consumption requires the approval of the client. The authorization (and its revocation) is managed directly through the Enedis client account or by using the electricity supplier as a proxy. They are kept for 24 months before being automatically deleted. Note that, in the rest of this document, and due to this deletion process, series are processed over different periods depending on when each study is performed. In this study, all data are processed on the Enedis computational infrastructure. We do not have access to any nominative information. Any request should be addressed to Enedis (see the footnote above).

3.1.2 Methodology

The statistical analyses made in this study follow the same procedure but vary according to the considered perimeter. A **perimeter** defines the dataset population: which meters are included.

For the *ISSDA* and *LONDON* datasets, the perimeter contains the whole cleaned dataset (without series containing empty values). For the *ENEDIS* dataset, a perimeter is a tuple defined by a geographical area and a profile. A **geographical area** is defined as a French administrative area. It ranges from a city to the whole country (metropolitan

1. `https://www.enedis.fr/donnees-personnelles`

France). A **profile** is a classification of individual meters made by Enedis [2]. Each meter is attributed to a profile based on contractual information and past consumption. For example, the perimeter (NAT; RES*) represents all the residential meters (RES*) in the whole of metropolitan France (NAT).

Table 3.1 provides the notations used in this chapter. All these notations are relative to a given perimeter. A glossary containing all the definitions is available in Table 3.1 and in Appendix A.

Notation	Description
Meter	A Point Of Measurement (POM) collecting the consumption value at regular timestamps.
Profile	Specific to the ENEDIS dataset. Contract type and consumption pattern associated with a meter.
Geographical area	Specific to the ENEDIS dataset. A geographical area. Either the whole of metropolitan France or an administrative department.
$\mathcal{T}$	Set of timestamps with $t \in \mathcal{T}$ being a single timestamp.
$\mathcal{W}$	Set of all possible consumption values within a perimeter. Example: $\mathcal{W} = \{0, 1, ..., 36000\}$ (in Watt W).
$\mathcal{S}$	Set of power consumption time series. $\mathcal{S}_{i,t}$ is the consumption of the i^{th} meter at the timestamp t. Note that for some timestamps, $\mathcal{S}_{i,t} = \varnothing$ and does not exist (ignored in the computation).
$\sigma(\mathcal{S}_t)$	Standard deviation for the set of series $\mathcal{S}$ at the timestamp t. Defined in Formulae 3.2.
$\rho(i, j)$	Pearson coefficient between the series i and j. Defined in Formulae 3.4.
$F[w]$	Frequency for the consumption value w. Defined in Formulae 3.5.
e_t	Shannon entropy. Defined in Formulae 3.8.
u_t^k	Uniqueness at timestamp t with k consecutive timestamps. Defined in Formulae 3.7.

Table 3.1 – Notations

Cleanup: We clean the raw datasets in the following manner. For the *ISSDA* and *LONDON* datasets, we remove all measurements containing missing values. For the *ENEDIS*

2. `https://www.enedis.fr/responsable-dequilibre-profilage-et-profils`

datasets, we selected the measurements with at least 46 measurements per day on the period. We detail any additional filters and missing values remediation methods applied when relevant.

Time series: A time series is a sequence of timestamped values. We represent (see eq. 3.1) a set of time series as a $|\mathcal{I}| \times |\mathcal{T}|$ matrix $\mathcal{S}$ where the $|\mathcal{I}|$ rows represent the time series of each individual and the $|\mathcal{T}|$ columns represent the timestamps (i.e., a sequence of integers from 1 to $|\mathcal{T}|$ for simplicity). Each cell $\mathcal{S}_{i,t} \in [d_{min}, d_{max}]$ thus represents the value of the i^{th} individual at the t^{th} timestamp. As no measurements could be negative, $d_{min} = 0$.

$$\mathcal{S} = \begin{bmatrix} \mathcal{S}_{1,1} & \cdots & \mathcal{S}_{1,|\mathcal{T}|} \\ \vdots & \vdots & \vdots \\ \mathcal{S}_{|\mathcal{I}|,1} & \cdots & \mathcal{S}_{|\mathcal{I}|,|\mathcal{T}|} \end{bmatrix} \tag{3.1}$$

Standard deviation: Equation 3.2 defines the standard deviation $\sigma(\mathcal{S}_t)$ of the set of time series $\mathcal{S}$ at the timestamp t. The standard deviation measures the dispersion of a set of the values relative to the mean value.

$$\forall t \in \mathcal{T}, \sigma(\mathcal{S}_t) = \sqrt{\frac{1}{|\mathcal{I}|} \cdot \sum_{\forall i \in \mathcal{I}} \left(\mathcal{S}_{i,t} - \overline{\mathcal{S}_t}\right)^2} \tag{3.2}$$

Covariance: Equation 3.3 defines the covariance of two time series i and j in the set of time series $\mathcal{S}$. The covariance measures the joint variation of the two series.

$$cov(i,j) = \frac{\sum_{\forall t \in |\mathcal{T}|} (\mathcal{S}_{i,t} - \overline{\mathcal{S}_i}) \cdot (\mathcal{S}_{j,t} - \overline{\mathcal{S}_j})}{|\mathcal{T}|} \tag{3.3}$$

Pearson coefficient: Equation 3.4 defines the Pearson coefficient. The Pearson coefficient is a metric measuring the correlation between two series (i and j in the set of time series $\mathcal{S}$). σ_i is the standard deviation of the series i (with respectively σ_j for series j). Equation 3.3 defines the covariance $cov(i,j)$.

$$\rho(i,j) = \frac{cov(i,j)}{\sigma_i \cdot \sigma_j} \tag{3.4}$$

Frequencies: Equation 3.5 defines the frequencies. Frequencies are computed for a set of time series $\mathcal{S}$. Each value of the series belongs to the definition domain $\mathcal{W} = \{d_{min} \cdots d_{max}\}$. $\delta(i,t,w) = 1$ when $\mathcal{S}_{i,t} = w$ and 0 otherwise.

$$\forall w \in \mathcal{W}, F[w] = \frac{\sum_{\forall t \in \mathcal{T}} \sum_{\forall i \in \mathcal{I}} \delta(i,t,w)}{\sum_{\forall w \in \mathcal{W}} \sum_{\forall t \in \mathcal{T}} \sum_{\forall i \in \mathcal{I}} \delta(i,t,w)} \tag{3.5}$$

Uniqueness: We apply the same method for computing the uniqueness regardless of the dataset. We compute the *ENEDIS* results using SQL and we developed a dedicated Python library [3] to compute the *ISSDA* and *LONDON* results.

Given k, we compute the uniqueness at each timestamp t by considering for each time series the sub-sequence starting at t (included) and containing k consecutive measures (called a *k-length sub-sequence* below) and by computing the uniqueness (i.e., number of unique measurements) in the resulting set of sub-sequences. Finally, for a given k, we compute and plot the average and minimum/maximum interval uniqueness over all timestamps.

Let S be a set of n electricity consumption time series with $|\mathcal{T}|$ timestamps each. Given k, we denote S_t^k the dataset derived from S that contains for each time series the k consecutive timestamps starting at time t (and consequently ending at time $t+k-1$). Note that we ignore the k-length sub-sequences that have missing values (due to, e.g., transmission errors). Missing values occur only on the *ENEDIS* datasets and concern up to 10 % of the measurements (see Section 3.4). The function $\mathtt{U}$, defined in Equation 3.6, outputs the set of time series that are **unique** in S_t^k.

$$s \in \mathtt{U}(S_t^k) \iff \nexists s' \in S_t^k \setminus \{s\}, \text{ s.t. } s' = s \tag{3.6}$$

The **uniqueness** at time t for k-length sub-sequences; denoted u_t^k and defined in Equation 3.7, is the fraction of unique time series given within the dataset S_t^k.

$$u_t^k = \frac{|\mathtt{U}(S_t^k)|}{|S_t^k|} \tag{3.7}$$

Entropy: The **entropy** is computed using the Shannon entropy formulae [Sha48] applied to k-length sub-sequences. Equation 3.8 computes the entropy for each timestamp

3. `https://gitlab.com/phd_antonin/tsunicity`

t. Equation 3.9 defines the probability to get the k-length sub-sequence $s \in S_t^k$ with $\delta(s, s') = 1$ if $s = s'$ and 0 otherwise.

$$e_t = -\sum_{s \in S_t^k} P[s] \log_2 P[s] \tag{3.8}$$

$$P[s] = \frac{\sum_{s' \in S_t^k} \delta(s, s')}{|S_t^k|} \tag{3.9}$$

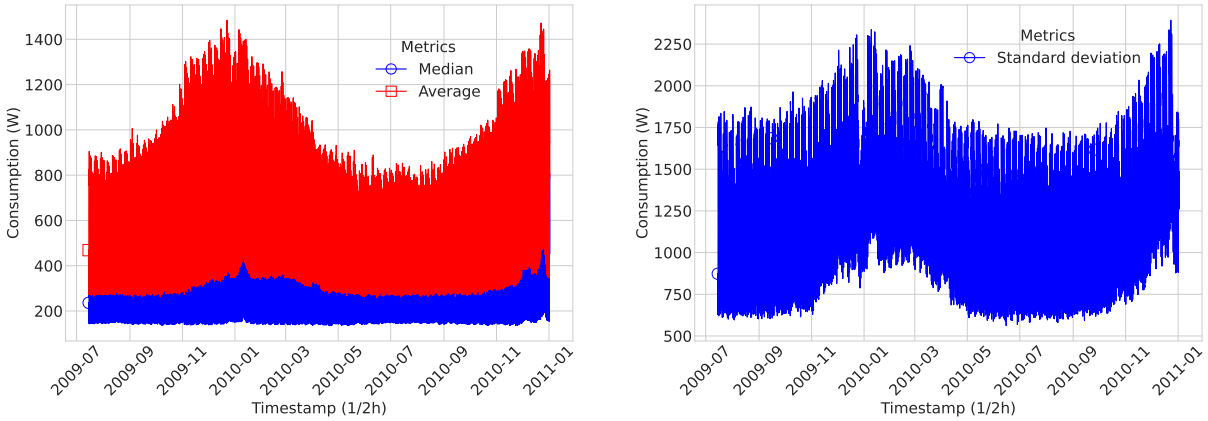
Rounding: We also study the impact of a more and more severe information loss on uniqueness. We degrade our time series by rounding their values by 1, 2, and 3 orders of magnitude (i.e., respectively to the closest 10 W (or Wh), the closest 100 W (or Wh), and the closest kW (or kWh)) and compute uniqueness and entropy as explained above. To this end, consumption measures are rounded before computing uniqueness or entropy.

A note on reading the figures: Due to the high number of figures, this study focuses on the most interesting figures. If there is no figure for a specific perimeter, refer to the analysis of the closest higher-level perimeter available. To keep the readability of the frequency figure, we display only a zoom of what is deemed the most interesting part of the figure. Note that this study has an appendix (Appendix B) containing all figures. The datasets contain measurement in local time: GMT for *ISSDA* and *LONDON* and CET for *ENEDIS*. It includes daylight saving time.

The boxplot figures, for example, Figure 3.16a, represent the Q25 (bottom of the box), Q75 (top of the box) and the median (middle line within the box). The whiskers are defined at $\pm 1.5 \cdot IQR$ (with $IQR = Q75 - Q25$). The points outside of the whiskers (for example, with Figure 3.16a in February) represent the outliers outside of that 1.5 IQR [4]. For the monthly statistics, January = 1 and December = 12. For the daily statistics, it is assumed that Monday = 0 and Sunday = 6.

On the uniqueness figures (for example, Figure 3.27), the line represents the average uniqueness, and the whiskers represent the minimum and maximum (respectively for the lower and upper whiskers) uniqueness measured.

4. `https://seaborn.pydata.org/tutorial/categorical.html#boxplots`



(a) Average and median consumption.

(b) Standard deviation

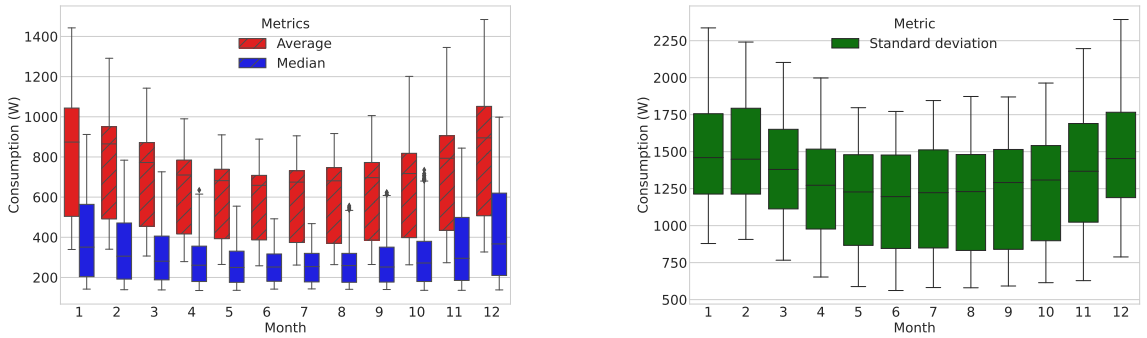
Figure 3.1 – Average, median consumption and standard deviation for the whole half-hourly *ISSDA* dataset.

3.2 ISSDA

The *CER-ISSDA* dataset [CER12] is an electric consumption dataset collected by the Commission for Energy Regulation for the Irish Social Science Data Archive. It contains the half-hourly electricity consumption measurements of 6435 Irish meters collected between July 2009 and December 2010. Besides, the dataset comes with a survey providing insight into the devices used by the meters (usage of heating devices) and socio-economic metrics such as the number of residents and employment status. We leave this survey apart from our studies.

Before using the data in our study, we cleaned the dataset by removing all measurements containing at least one missing value. This left us with 4622 complete half-hourly series over one year and a half. We changed the unit from kilowatt to Watt to manipulate only integer values. From now on, any mention of the *ISSDA* dataset refers to this cleaned dataset. We aggregated the values of the cleaned half-hourly dataset per day to get the daily consumption measurements. We refer to this dataset as *ISSDA-DAILY*.

General Statistics: Figure 3.1 shows the average, median, and standard deviation every half-hour. The figure shows that the average consumption is well above the median consumption and that the standard deviation is correlated to the average consumption. The figure also reveals two seasonal patterns: a first daily pattern, with substantial variation between the time of the day, and a second one depending on the season, with higher



(a) Average and median consumption

(b) Standard deviation

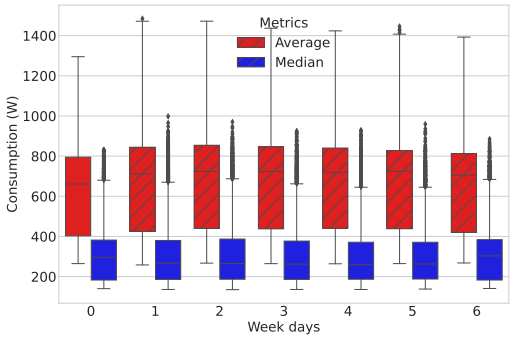
Figure 3.2 – Average, median consumption and standard deviation for the whole half-hourly *ISSDA* dataset (1 = January, 12 = December).

consumption during winter than during summer. Computing the Pearson coefficient between the average consumption and the standard deviation shows a strong correlation ($\rho = 0.82$), meaning we can expect more dispersed consumption when the average consumption is high.

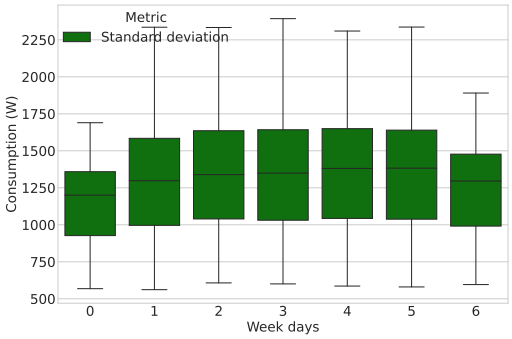
Monthly Statistics: The monthly statistics are illustrated by Figure 3.2. It shows a seasonal pattern with higher consumptions in winter than in summer. The standard deviation follows the consumption with more dispersed consumptions during the winter. The lower bound of the median indicates a fairly stable Q25 consumption over the year seemingly unaffected by the month.

Daily Statistics: Figure 3.3 shows the average and median consumptions and the standard deviation per day of the week (0 = Monday and 6 = Sunday). The consumption and the standard deviation are relatively stable during the week, with two notable exceptions: Monday and Sunday. These days, consumption, especially the Q75, is globally lower (around 20 %) than during the rest of the week.

Hourly Statistics: Figure 3.4 shows the average, median, and standard deviation per hour of the day. It shows a strong pattern with the lowest consumption during the night (0h to 6h) and the highest consumption during the evening (17h to 20h), with a high plateau during the day (9h to 17h). The standard deviation peaks around 11h and decreases

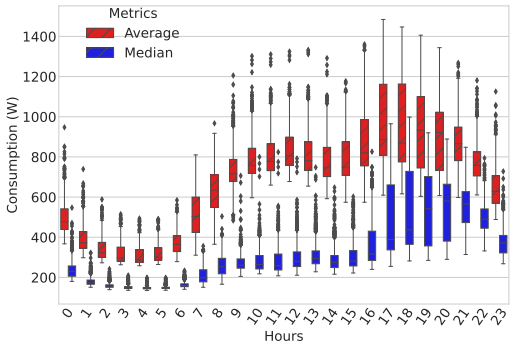


(a) Average and median consumption

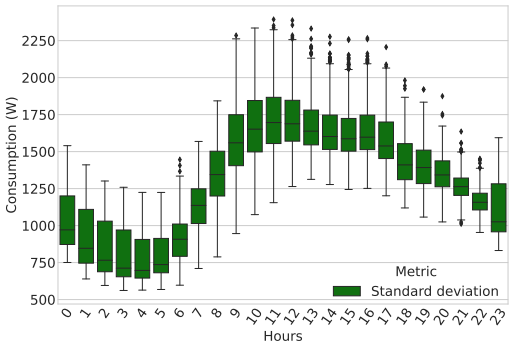


(b) Standard deviation

Figure 3.3 – Average, median consumption and standard deviation for the whole half-hourly *ISSDA* dataset (0 = Monday, 6 = Sunday).

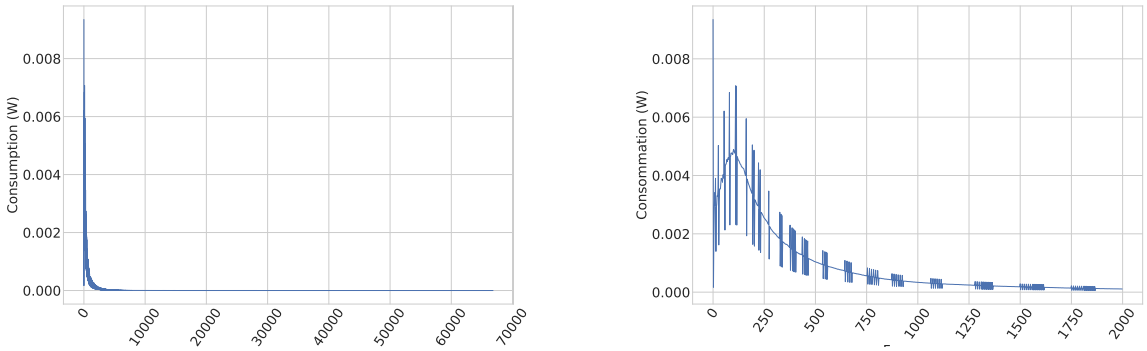


(a) Average and median consumption



(b) Standard deviation

Figure 3.4 – Average, median consumption and standard deviation for the whole half-hourly *ISSDA* dataset.



(a) [0; 69000] W window (b) [0; 2000] W window

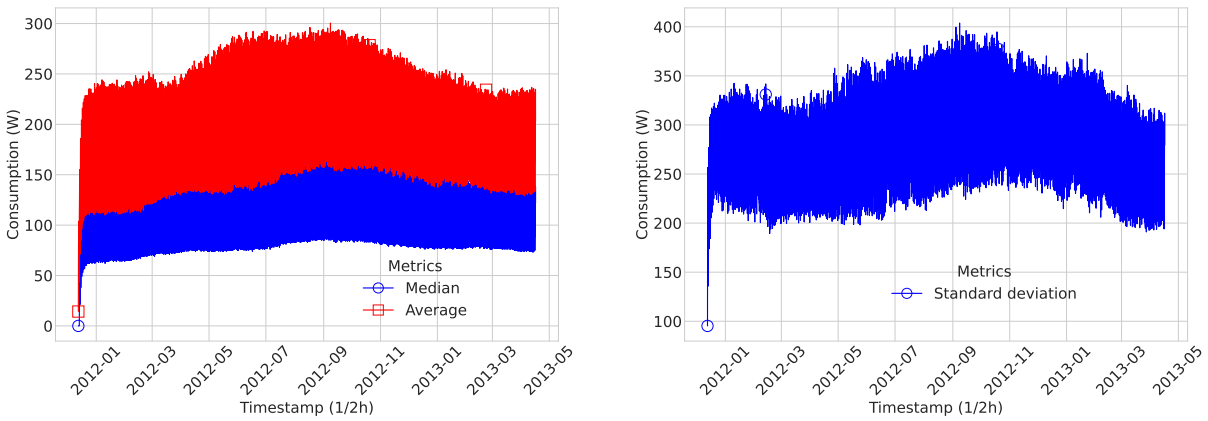
Figure 3.5 – *ISSDA* consumption frequency with a zoom on specific ranges.

afterward, ignoring the consumption peak of 17h showing more diverse consumption in the morning.

Consumption frequencies: Figure 3.5 shows the consumption frequency per Watt independently of the timestamps. Figure 3.5b zooms on frequencies below 2 kW. First, even if the consumption range is large (up to 69 kW), most recorded consumption happened below 1 kW with two peaks at 0 and 100 W. The figure displays some regular artifacts consisting of higher frequencies followed shortly after by lower frequencies. This is probably due to some rounding effect on the meter: higher frequencies are even values while lower frequencies are odd. This is not systematic and does not significantly influence the series' shape.

3.3 London

The *UK Power Networks* company handling the electricity distribution in and around London collected the *LONDON* dataset [Net13]. It contains the half-hourly measurements of 5567 meters collected between November 2011 and February 2014. A significant portion of the dataset population (approx. 1100 series) is subject to dynamic pricing depending on the time. The company applies financial incentives to shift their consumption to another time. The *UK Power Networks* company issues high price periods one day before the predicted consumption peak. High prices are mainly issued in the winter and dur-



(a) Average and median consumption

(b) Standard deviation

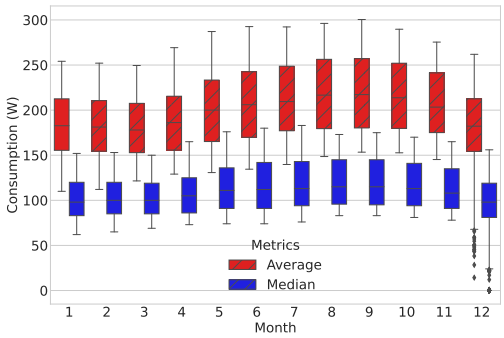
Figure 3.6 – Average, median consumption and standard deviation for the whole half-hourly *LONDON* dataset.

ing the evening [5]. Conversely, the company issues lower prices when they expect a low consumption period.

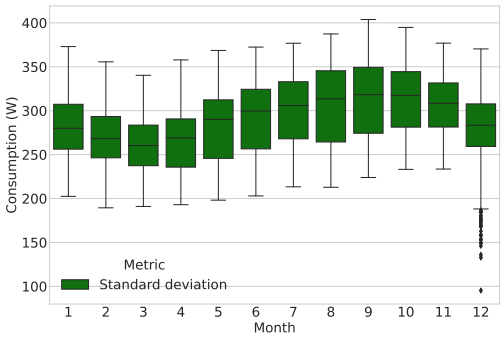
We clean the dataset by removing all measurements containing at least one missing value leaving us with 5433 complete series. We change the unit from kilowatt to Watt to manipulate only integer values. From now on, any mention of the *LONDON* dataset refers to the cleaned version of this dataset.

General Statistics: Figure 3.6 shows the average, median, and standard deviation every half-hour. The figures show that the average consumptions are well above the median consumption and that the standard deviation is correlated to the average consumption. Besides, they reveal two seasonal patterns: A first daily pattern, with strong variation depending on the time of the day (more on that in the hourly statistics paragraph), and a second seasonal pattern, with higher consumption during summer and autumn than winter. We note a low consumption on all parameters in the first days of the study due to a high proportion of consumptions at 0 W. 94 % meters in the first timestamp recorded a 0 W measurement. Computing the Pearson coefficient between the average consumption and the standard deviation shows a strong correlation ($\rho = 0.92$), meaning we can expect more dispersed consumption when the average consumption is high.

5. `https://innovation.ukpowernetworks.co.uk/projects/low-carbon-london/`

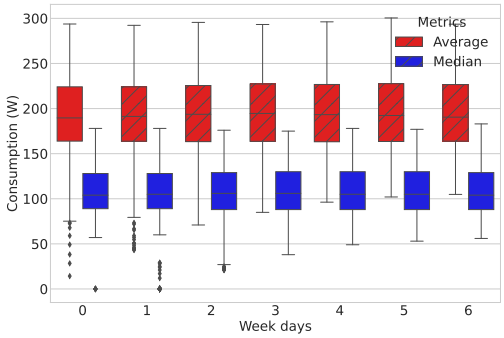


(a) Average and median consumption

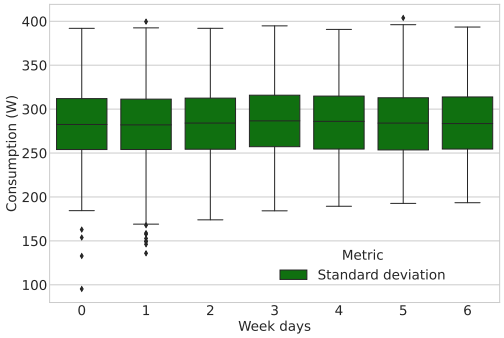


(b) Standard deviation

Figure 3.7 – Average, median consumption and standard deviation for the whole half-hourly *LONDON* dataset(1 = January, 12 = December).



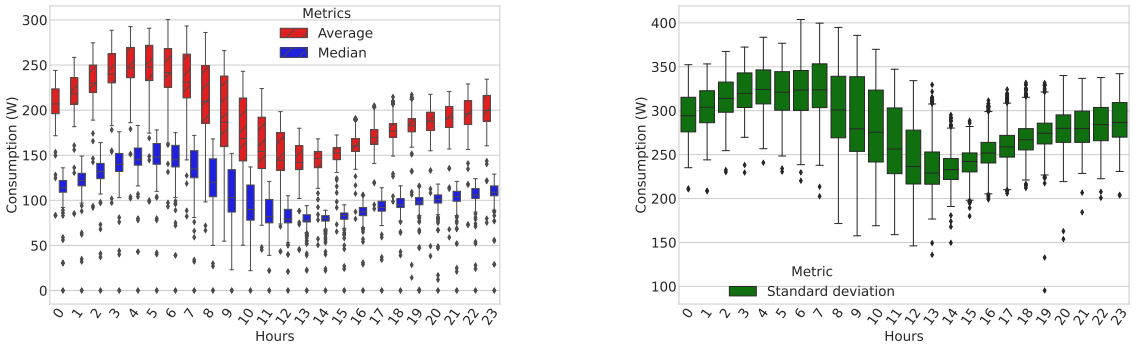
(a) Average and median consumption



(b) Standard deviation

Figure 3.8 – Average, median consumption and standard deviation for the whole half-hourly *LONDON* dataset (0 = Monday, 6 = Sunday).

Monthly Statistics: Figure 3.7 illustrates the monthly statistics. We remark on a surprising seasonal pattern with higher summer consumption than winter. This is surprising compared to the *ISSDA* (Section 3.2) and *ENEDIS* (Section 3.4.1) dataset. They both have stronger consumption in winter than in summer. We could have expected higher consumption due to the generally lower temperature in winter. As the dataset measures a lower power than for their *ISSDA* and *ENEDIS* counterparts (approx. twice less), we guess the households do not use electricity for heating. The standard deviation follows the consumption with more dispersed consumption during the autumn.



(a) Average and median consumption

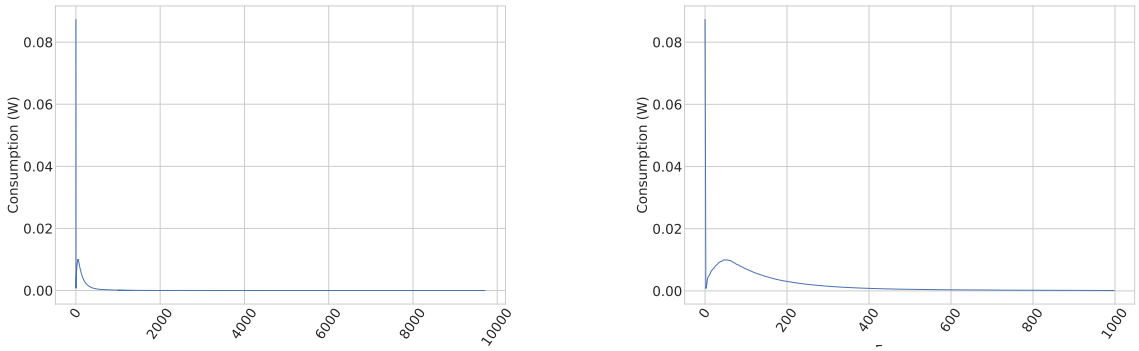
(b) Standard deviation

Figure 3.9 – Average, median consumption and standard deviation for the whole half-hourly *LONDON* dataset.

Daily Statistics: Figure 3.8 shows the average and median consumptions and the standard deviation per weekday (0 = Monday and 6 = Sunday). They are both relatively stable during the week without any days standing to the other.

Hourly Statistics: Figure 3.9 shows the average, median, and standard deviation per hour of the day. It shows a strong pattern, with the highest consumption happening at night (3h to 8h) and the lowest around noon (11h to 14h). This pattern is surprising compared to the *ISSDA* (Section 3.2) and *ENEDIS* (Section 3.4.1), which have the lowest consumptions during the night and peaking during the evening. This could be due to the different pricing policies encouraging some individuals to shift their consumption at different times of the day.

Consumption frequencies: Figure 3.10 shows the consumption frequency per Watt independently of the timestamps, and Figure 3.10b zoom on the frequencies below 1 kW. While *ISSDA* and *ENEDIS* have a maximum recorded consumption of respectively 69 kW and 36 kW, the *LONDON* dataset has a maximum recorded consumption below 10 kW. As for the other dataset, most recorded consumption happened below 1 kW with two peaks at 0 (up to 9 % of the measurements) and 50 W (up to 1 % of the measurements).



(a) [0; 69000] W window (b) [0; 2000] W window

Figure 3.10 – *LONDON* dataset consumption frequency (with zoom).

3.4 Enedis

The *ENEDIS* dataset used in this study was collected by ENEDIS between September 2020 and September 2021. We focus on metropolitan French residential and professional meters and split them into several smaller perimeters per profile and geographical areas. As a disclaimer, please note that the COVID-19 pandemic impacted the studied period and might not fully represent a typical year [6]. It was chosen as the newest during the study and with the most measurements available before the GDPR purges. The GDPR purges delete most measurements two years after their creation (see Section 3.1.1). Besides, even if around 3 million meters collect half-hourly series, they represent only a portion of the 35 million meters installed by ENEDIS. The other meters do not have the half-hourly measurements activated.

Perimeters: This study focuses on the entire French metropolitan (National/NAT) territory and a few selected regions (at the French departmental administrative level). Perimeters are summarized in Table 3.2. Departments are chosen based on their (assumed) geographical specificity that might impact consumption (climate, big cities). The Paris (75) region corresponds to a dense urban area, Haute-Savoie (74) is a mountainous region, Marne (51) is a rural region with a continental climate, Finistère (29) is an oceanic region, and Bouches-du-Rhône (13) is a Mediterranean region with a big city (Marseille) and heavy industries.

6. `https://www.enedis.fr/presse/quel-impact-eu-le-confinement-sur-votre-consommation-delectricite`

The profiles are ENEDIS's most common residential (RES1, RES2) and professional (PRO1, PRO2) commercial contracts. We added a few less common contracts deemed interesting for this study: RES11, PRO5. The remaining residential and professional contracts are grouped in the RESAutre and PROAutre profiles. The RES1 and PRO1 represent contracts that always have the same pricing (BASE). The RES2 and PRO2 contracts represent contracts with different pricing depending on the time of the day (HPHC). The period impacting the pricing is different for each individual. RES11 profiles are high consumers (authorized by their contract) BASE residential meters, while the RES1 profile contains only low consumption meters. The PRO5 profile contains public lighting meters. The RESAutre and PROAutre profiles contain a melting pot of smaller niche profiles. We do not study them individually here.

Notation	Description
Profiles:	
RES1	BASE residential meters (same price every time) ≤ 6 kW.
RES2	HPHC residential meters (price variation depending on time).
RES11	BASE residential meters (same price every time) > 6 kW.
RESAutre	mixing of all other residential profiles.
RES*	RES1 $\cup$ RES2 $\cup$ RES11 $\cup$ RESAutre.
PRO1	BASE professional meters (same price every time).
PRO2	HPHC professional meters (price variation depending on time).
PRO5	Public lighting.
PROAutre	Mixing of all other professional profiles.
PRO*	PRO1 $\cup$ PRO2 $\cup$ PRO5 $\cup$ PROAutre.
RES*+PRO*	RES* $\cup$ PRO*.
Geographical areas:	
NAT	Contains all of metropolitan France.
13	The Mediterranean department "Bouche-du-Rhône" (13).
29	The oceanic department "Finistère" (29).
51	The continental department "Marne" (51).
75	The alpine department of "Haute-Savoie" (74).
75	The department "Paris" (75).

Table 3.2 – ENEDIS perimeters

3.4.1 General statistics

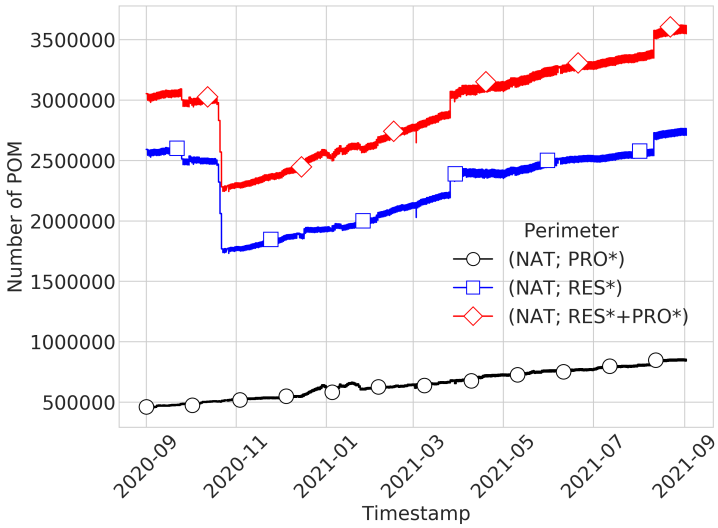


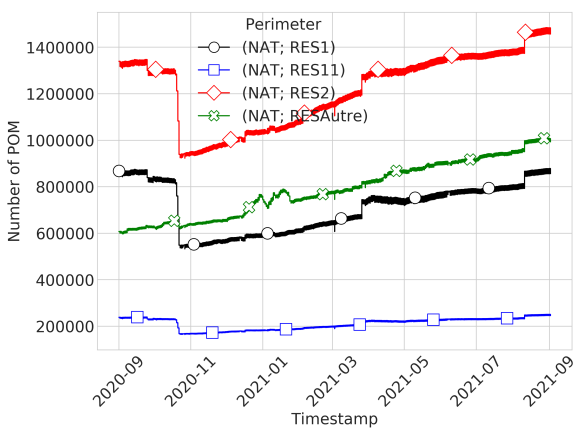
Figure 3.11 – Number of meters (POM) per 1/2h for the perimeters : [(NAT; RES*+PRO*), (NAT; RES*), (NAT; PRO*)]. In total there is approximately 3M POM with 2.5M RES* and 1M PRO* POM.

Number of meters: The number of meters (POM) corresponds to the number of meters that activated the half-hourly measurement and have been successfully reached at the timestamp. This metric gives an idea of the volume of measurements involved in the rest of this section.

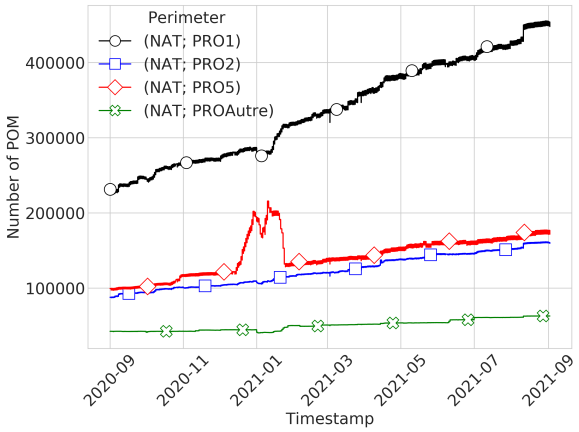
Figure 3.11 shows the number of POM available in France without profile discrimination and with RES* and PRO* discrimination. There are approximately 3 million (NAT; RES*+PRO*) measurements per timestamp. There are approximately 5 times more (NAT; RES*) meters (around 2.5M) than (NAT; PRO*) meters (around 500k).

Figures 3.12 show the number of POM available for each RES and PRO subdivision. For the RES meters, the (NAT; RES2) meters are the more numerous (around 1.2M), followed by the (NAT; RESAutre) meters (up to 1M meters) and the (NAT; RES1) meters (up to 1M). The (NAT; RES11) meters are relatively stable, with around 250K meters. The (NAT; PRO1) are the more numerous for the PRO-meters, with up to 400k meters. The (NAT; PRO5) and (NAT; PRO2) are relatively similar to both, around 100k meters. The (NAT; PRO5) are slightly more numerous than the (NAT; PRO2). The (NAT; PROAutre) profile contains around 50K meters.

The drop in October 2020 may be due to the removal of a large number of meters forcibly (at least without their opt-in approval) registered by their providers to the half-hourly metering program. The peak in the number of POM of the (NAT; PRO5) in

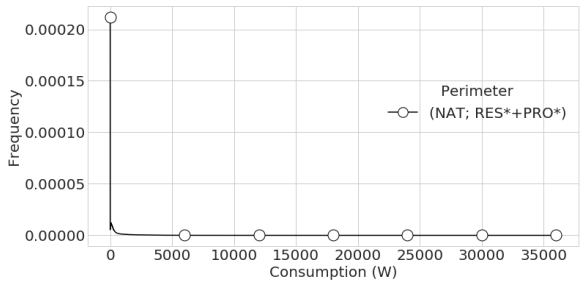


(a) Perimeters: [(NAT; RES1), (NAT; RES2), (NAT; RES11), (NAT; RESAutre)]

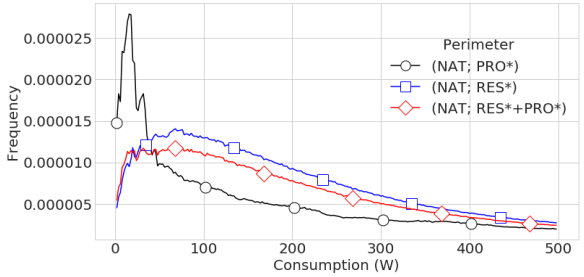


(b) Perimeters: [(NAT; PRO1), (NAT, PRO2), (NAT, PRO5), (NAT, PROAutre)]

Figure 3.12 – Number of POM per 1/2h for the subdivision of *ENEDIS* National RES and PRO profiles at 1/2h step. The number of POM increase over time with some temporal increase/drop in the number of POM.



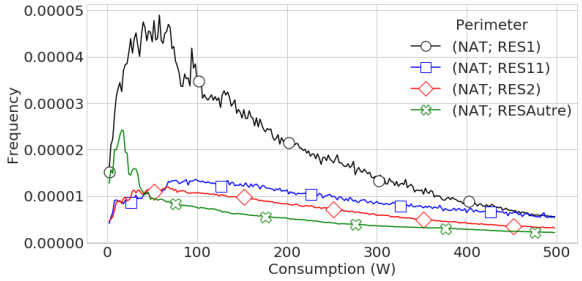
(a) Perimeter: (NAT; RES*+PRO*)



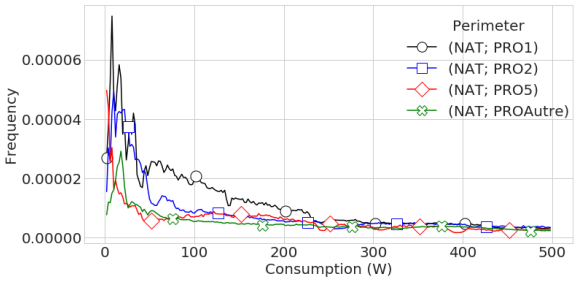
(b) Perimeters: [(NAT; RES*+PRO*), (NAT; RES*), (NAT; PRO*)]. Windows: [1; 500]W.

Figure 3.13 – *ENEDIS* consumption frequency.

December 2020 / January 2021 may be due to the addition/removal of Christmas lighting (and their associated meters) or a punctual measurement campaign. The temporary measurement campaign may also happen elsewhere in the figures (such as the drop in October 2020). Some small and local drops (such as in March 2021) may be due to bugs during the collection process. However, we note that these interpretations are educated guesses by ENEDIS colleagues and may not be the real reason as no absolute information can be found. All figures show that the number of measurements linearly increases with time and the deployment of new meters.



(a) Perimeters: [(NAT; RES1), (NAT; RES2), (NAT; RES11), (NAT; RESAutre)]



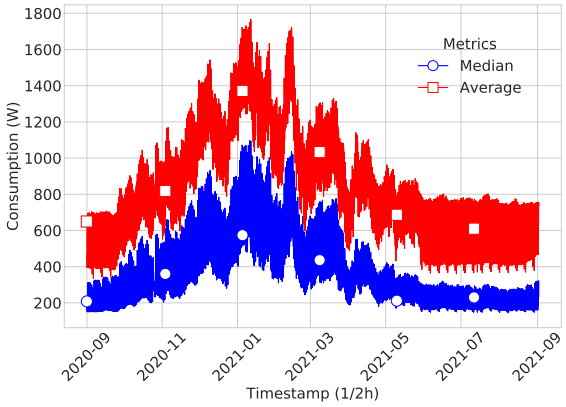
(b) Perimeters: [(NAT; PRO1), (NAT; PRO2), (NAT; PRO5), (NAT; PROAutre)]

Figure 3.14 – *ENEDIS* consumption frequency ([1; 500]W window).

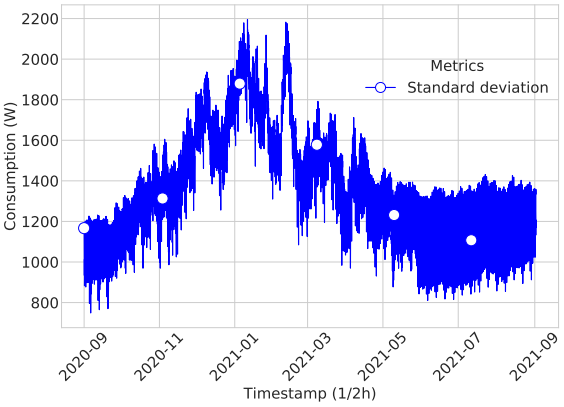
Consumption frequency: In the following, we present the consumption frequencies independently of the timestamps. The objective is to learn when the consumptions happened and how they distribute.

Figures 3.13 and 3.14 shows the consumption distribution frequency. Figure 3.13a shows all the frequencies without any filter, while the other figures show a zoom of the [1; 500] W window. It shows that the consumption at 0 is overly represented (around 2 % of the total measurements) and quickly drops for higher consumption. The over-representation of the consumptions at zero is for all the perimeters considered and not only for higher consumptions. All other figures focus on a window of consumption between 1 and 500 W. Figure 3.13b shows the consumptions frequencies of the (NAT; RES*+PRO*), (NAT; RES*) and (NAT; PRO*) perimeters. It shows that the global frequencies distribution (NAT; RES*+PRO*) closely follows the distribution of the (NAT; RES*). Figure 3.14a shows the distribution of each RES profile, and Figure 3.14b shows the distribution of each PRO* profile.

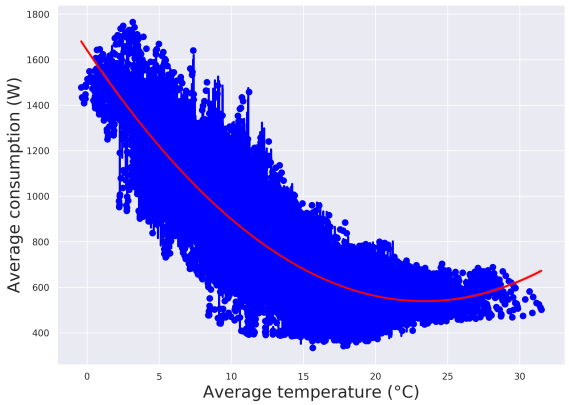
All figures, except the PRO5, follow a similar pattern: a low frequency at low consumptions that increases with the consumption until the frequency decreases until the maximum consumption. However, the peak's location and height vary according to the profile. This is particularly true for the RES profiles, as the PRO frequencies are closer to each other. The PRO5 have higher frequencies at lower consumptions that quickly decrease, followed by a slight bounce around 150 W. The null consumption is overly represented, and the consumptions are concentrated on the lower values. However, all the possible consumptions are represented even at low frequencies.



(a) Average and median per half-hour



(b) Standard deviation per half-hour



(c) *ENEDIS* dataset thermosensitivity: impact of the average consumption depending on the average temperature.

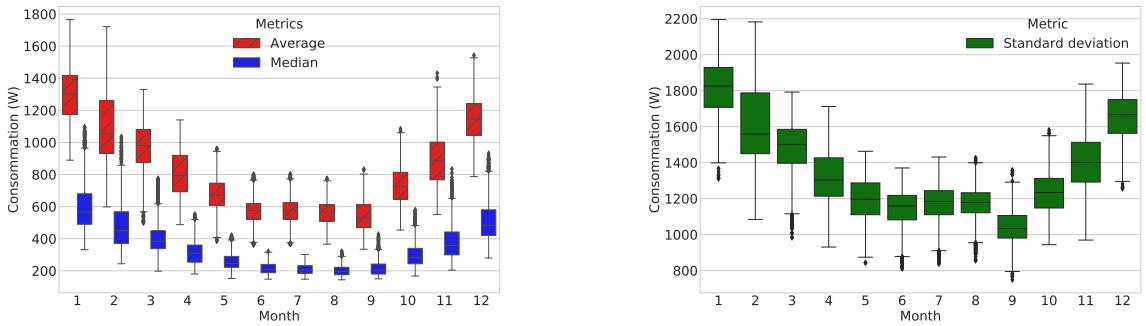
Figure 3.15 – Perimeter: (NAT; RES*+PRO*)

3.4.2 The general case

This section presents various statistics over the (NAT; RES*+PRO*) perimeter containing all the measurements. We use it as a base analysis by considering the statistics obtained by this perimeter as the normal case. In the following sections, we use this general case as a comparison base to present individual perimeters specificities standing out from this general case.

Half hourly consumption and thermosensitivity: Figure 3.15 shows the average consumption and standard deviation every half-hour and the thermosensitivity. The average consumption is almost always above the median consumption, and the standard deviation is correlated to the average consumption. Computing a Pearson coefficient be-

tween the standard deviation and the quantile 25 ($\rho = 0.62$), average ($\rho = 0.95$), and quantile 95 ($\rho = 0.97$) half-hourly series shows a strong correlation between consumption level and the standard deviation. With a Pearson coefficient of 0.95, the standard deviation is strongly correlated with the average consumption. On both figures (Figure 3.15a and Figure 3.15b) we can easily distinguish two seasonal areas: the summer, where the consumption is relatively stable (within the daily variation), and the winter, where the consumption is higher with higher volatility linked to the temperature. Temperatures negatively correlate with consumption (Pearson coefficient at -0.67). When the temperatures drop below 15°C, the consumption increases. The impact of the temperature on the consumption is confirmed by Figure 3.15c. This figure shows the thermosensitivity: the impact of the temperature on the average consumption. It displays the average consumption according to the average national temperature (each point is a half-hourly average consumption) and a polynomial regression (order 2) on those points. It helps visualizing the correlation between the temperature and the consumption when the temperature drops below 15°C.

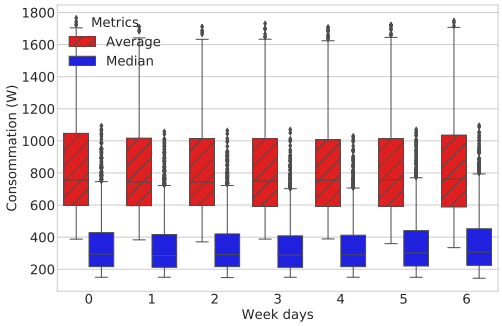


(a) Average and median per calendar month

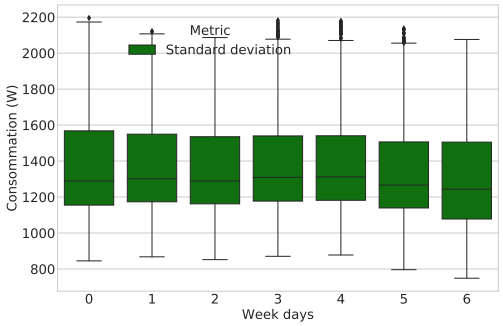
(b) Standard deviation per calendar month

Figure 3.16 – Perimeter : (NAT; RES*+PRO*).

Monthly statistics: The monthly statistics illustrated by Figure 3.16 confirm the consumption seasonality except for a slight bounce in July. On the other hand, the standard deviation seems to be less impacted by the temperature. The drop between August (8) and September (9) is linked to running our experiments from September 2020 to September 2021. Therefore there is one year of difference between these two points. The half-hourly Figure 3.16b indeed shows a greater dispersion during the end of the 2021 period than in September 2020.

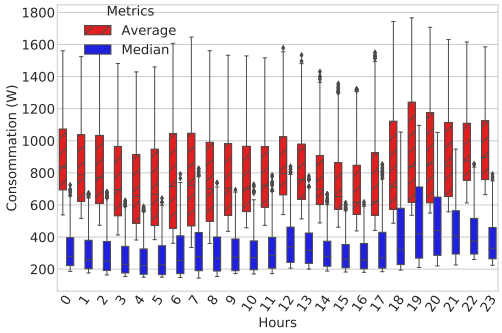


(a) Average and median per week days

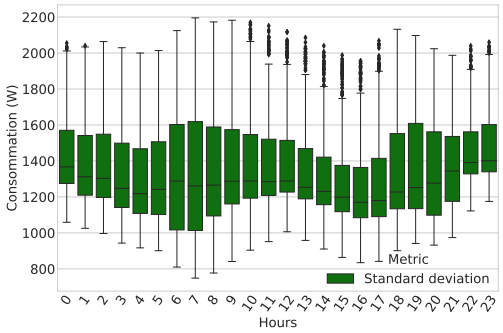


(b) Standard deviation per week days

Figure 3.17 – Perimeter : (NAT; RES*+PRO*)



(a) Average and median per hours of the day

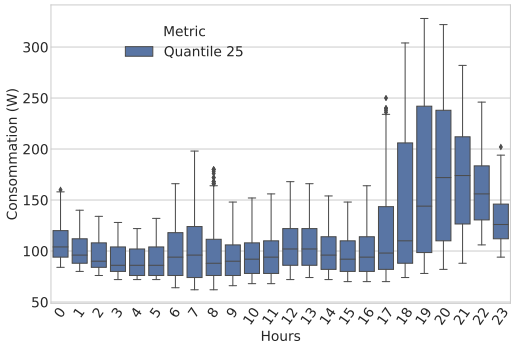


(b) Standard deviation per hours of the day

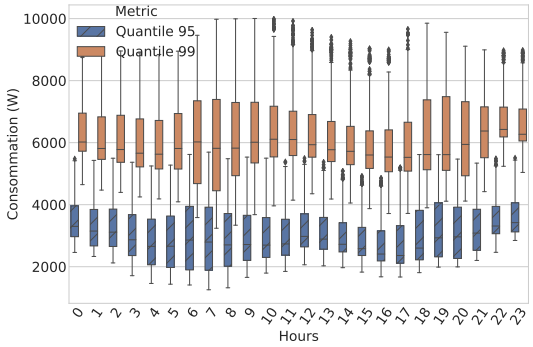
Figure 3.18 – Perimeter : (NAT; RES*+PRO*)

Daily statistics: Figure 3.17 shows the variation of the average, the median (Figure 3.17a) and the standard deviation (Figure 3.17b) per week days as a boxplot. Our initial intuition was that the consumption varies over the days, with at least a distinction between the working days and the weekends. However, the figures show high stability with very few differences from one day to another.

Hourly statistics: Figure 3.18 shows the variation of the average, the median, and the standard deviation per hour of the day. Even if the average consumption seems stable (with slight peaks in the morning and the evening), the median shows higher consumption during the day, with a strong consumption peak in the evening and a smaller peak in the morning. The standard deviation (Figure 3.18b) shows a greater dispersion during the day than at night. In contrast with the median, the stronger dispersion appears in the



(a) Quantile Q25 per hour of the day.



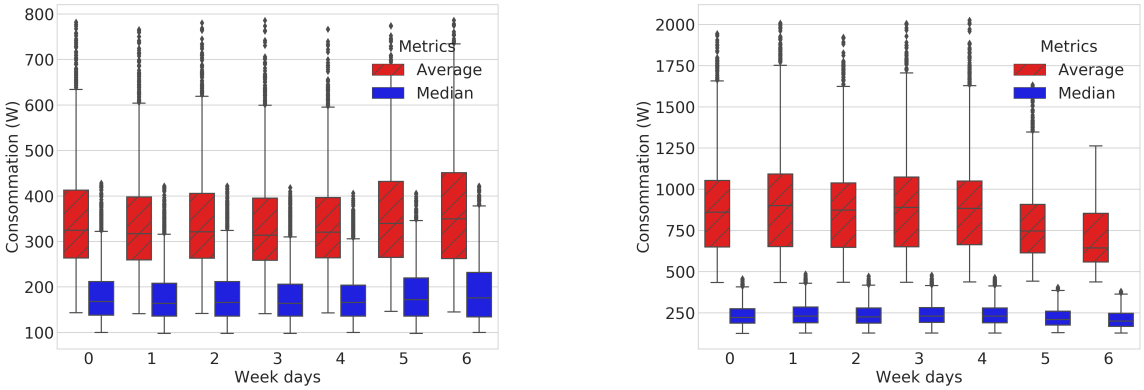
(b) Quantile Q95 and Q99 per hour of the day.

Figure 3.19 – Perimeter : (NAT; RES*+PRO*)

morning rather than in the evening (although they still have a higher dispersion than the rest of the day). Although the consumptions are higher in the evening, they are more diverse in the morning.

Figure 3.19 shows the hourly variation of the Q25 (Figure 3.19a), Q95 and Q99 (Figure 3.19b). They highlight the atypical lower (respectively higher) consumption. Figure 3.19a shows a very strong peak in the evening and a more stable consumption for the rest of the day. Higher quantiles (in Figure 3.19b) are stable with still a higher consumption in the morning and in the evening than during the rest of the day. However, the most notable feature is the huge consumption gap between Q25, the median, and the Q95 department, with the Q25 consumption being around 5 times lower than the median consumption and the Q95 consumption being 8 times higher than the median. The highest difference is obtained by Q99 in the evening consuming between 16 and 20 times more than the median.

Synthesis of the general case: To conclude this section, we can define our consumptions as mainly low (< 1 kW) with some relatively rare outliers with huge consumptions. The average consumption is around 500 W. They are strongly correlated with the temperatures leading to a seasonal consumption pattern. They have higher consumption during the winter than during the summer. Surprisingly, this perimeter does not display noticeable consumption changes between weekdays. On the intra-day, consumptions are greater and more volatile during the day than at night. We note some exceptions depending on the



(a) Perimeter: (NAT; RES1). (b) Perimeter: (NAT; PRO1).

Figure 3.20 – Average and median consumption per day of the week on the *ENEDIS* dataset.

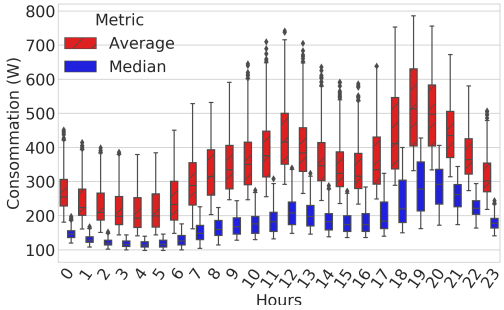
quantiles observed. On every level observed, the standard deviation follows the average consumption leading to a stronger dispersion for higher consumption.

In the following section, we show that each profile has its characteristics. We study these various profiles and their differences from the general case.

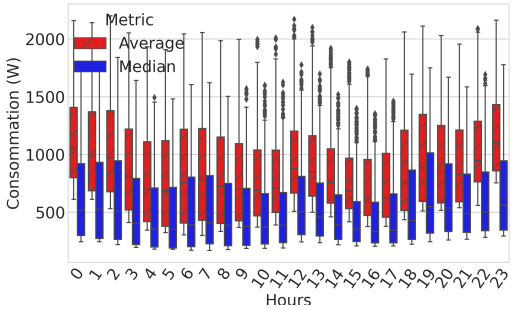
3.4.3 Divergences from the general case

In the previous section, we described our consumptions with global statistics, and we defined a statistical point of reference built from the (NAT; RES*+PRO*) perimeter containing all the measurements. However, each sub-perimeter, geographical, or profile has its specificity. This section focuses on the differences from the general case appearing on the sub-perimeter basis. First, we present the primary differences between RES, PRO, and their main sub-profiles. Then, we focus on some outliers profiles that have unusual consumption properties. We only put the dispersion figures for readability when they differ from the conclusions drawn from the average and median consumption figures. For further analysis, please refer to Appendix B.

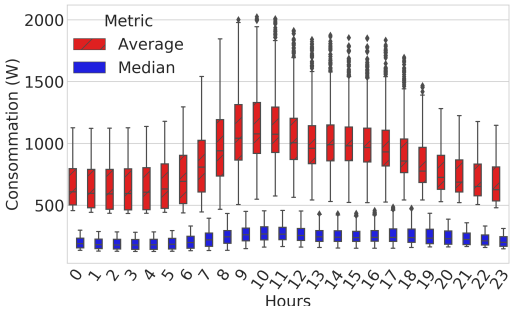
Daily differences between RES and PRO: Surprisingly, the general statistics show little to no difference between the days of the week. However, zooming into the RES* and PRO* sub-profiles (Figure 3.20) highlights some differences. We see that the RES meters consume more during the weekend than during the working days and that the



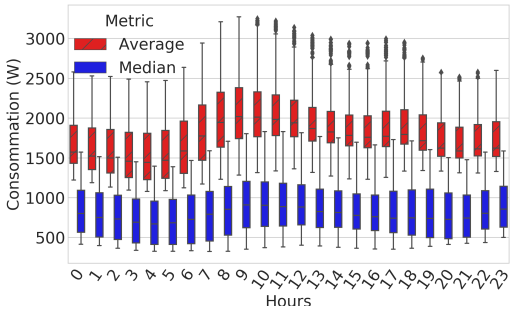
(a) Perimeter: (NAT; RES1).



(b) Perimeter: (NAT; RES2).



(c) Perimeter: (NAT; PRO1).

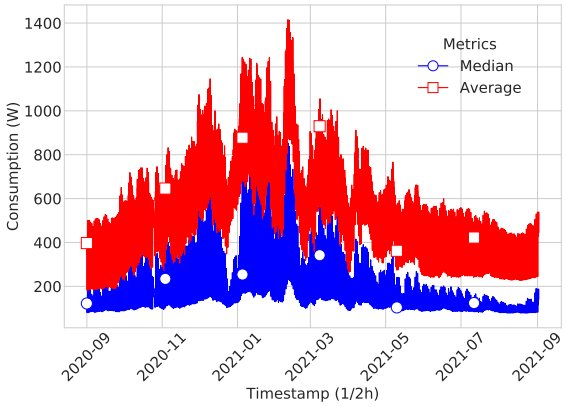


(d) Perimeter: (NAT; PRO2).

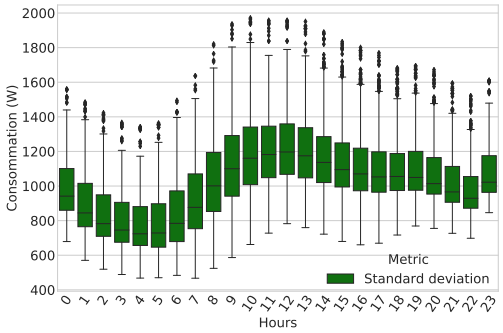
Figure 3.21 – Average and median per hours of the day on the *ENEDIS* dataset.

PRO meters consume less during the weekend. These changes appear on most individual profiles except for the PRO5 meters.

Main profiles hourly differences: The main difference between RES and PRO profiles comes with the different consumption values. Generally, the PRO* profiles consume more (average of around 1200 W) than the RES* profiles (average of around 750 W). However, the hourly consumptions (Figure 3.21) show more significant differences between profile consumptions. The RES1 profiles have two consumption peaks in the morning and the evening (higher). They consume 350 W on average. This pattern is shared by the RES11 meters with smoother peaks and much higher average consumptions at 750 W. On the other hand, the RES2 consumes, on average more than the RES1 and RES11 at 1000 W but has a much more erratic hourly consumption pattern. The PRO1 and PRO2 follow a similar consumption pattern with a peak in the morning (between 8h and 12h), less consumption in the afternoon, and lower consumption at night. The PRO2 hourly consumption pattern is smoother than the PRO1, with a less notable day-night differ-



(a) Average and median consumption per 1/2h.



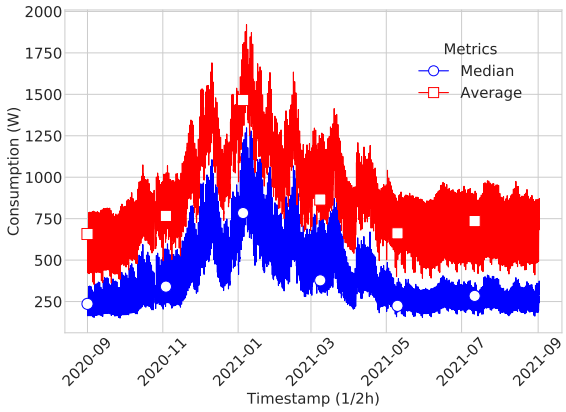
(b) Hourly standard deviation.

Figure 3.22 – *ENEDIS* perimeter: (75; RES*+PRO*)

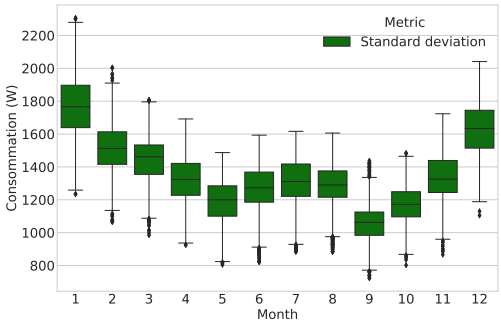
ence. On average, the PRO2 consumes twice as much (1500 W) as the PRO1 (750 W). The PRO profiles are more dispersed (2100 W on average for the PRO*) than the RES profiles (948 W on average). The most dispersed profiles are the PROAutre (average at 2700 W) followed by the PRO2 (2590 W), RESAutre (2000 W), PRO5 (1690 W), and PRO1 (1680 W). The least dispersed profiles are the RES2 (1000 W), RES11 (1000 W), and RES1 (488 W). These results confirm that the dispersion seems correlated with the average consumption, with higher consumption leading to higher dispersion.

Regional sensibility: Most studied regions follow a similar consumption pattern to the general case except for the Paris region (75).

In Paris, Figure 3.22a shows strong consumption drops around Christmas and lower consumption during the summer holidays (July-August). On the hourly level (Figure 3.22b), consumption follows a (NAT; PRO1) pattern: higher consumption during the day than during the night with a consumption peak in the morning. In Paris, the RES1 and PRO1 are more numerous than the national level at the detriment of the RES2, PRO2, and PRO5. On 2021-08-31, there is approximately 90,000 (75; RES*+PRO*) POM in Paris and 3,510,000 (NAT; RES*+PRO*) POM on the national level. There is twice the proportion of RES1 in Paris compared to the NAT level. There is 45 % (approximately 40,500) RES1 in Paris against 24 % (approximately 870,000) at the NAT level. The proportion of RES2 is half of the national level, with 24 % (approximately 22,000) against 41 % (approximately 1,455,000). With 16 % (approximately 15,000) against 11 % (approximately



(a) Average and median consumption per 1/2h.



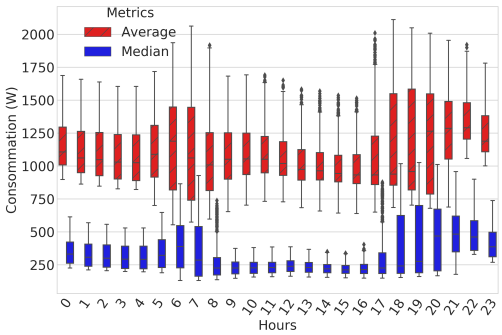
(b) Hourly standard deviation.

Figure 3.23 – *ENEDIS* perimeter: (13; RES*+PRO*)

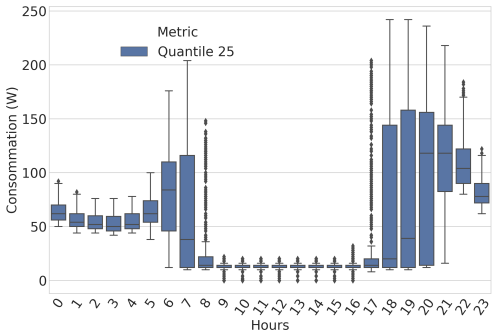
418,000), the proportion of PRO1 is slightly up. There is slightly less PRO2 in Paris, with 3.6 % (approximately 3,200) against 4.23 % (approximately 150,000). PRO5 profile is marginalized with only 0.4 % (41) against 4.83 % (approximately 170,000) and the proportion of RESAutre (1 % -900 POM- against 4 % -140,000 POM-). The proportion of RES11 and PROAutre are stable with 7.49 % (6,700 POM) against 6.96 % (244,000 POM) and 1.35 % (1,210 POM) against 1.63 % (57,000 POM) for the PROAutre.

The Bouche-du-Rhône region (13) shows some consumptions bouncing during the summer, with Figure 3.23 showing these bounces as smaller sporadic peaks in the summer. On the 2021-08-31, there is approximately 115,000 (13; RES*+PRO*) POM and 3,510,000 (NAT; RES*+PRO*) POM on the national level. In proportion, there are slightly fewer RES1, RES11, PRO5, and PROAutre with respectively 24,000 (21 %), 5,700 (5 %), 3,888 (3.38 %), and 1,260 (1.10 %) POM each. There are slightly more RESAutre with 4.72% (5,400 POM) against 4 % and PRO2 4.62 % (5,300 POM) against 4.23 %. The proportion of PRO1 is equivalent to 12.26 % (14,000 POM) against 11.93 %.

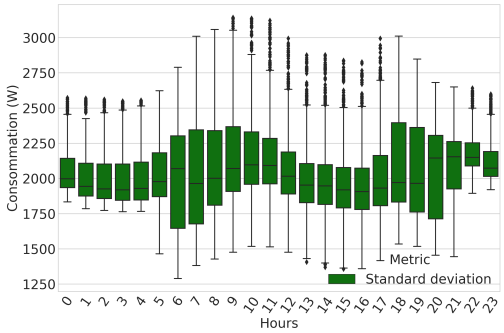
Generally, the proportion of profiles is closer in the Bouche-du-Rhône region to the general case than in Paris. As for the average consumption and standard deviation, the (NAT; RES*+PRO*), (29; RES*+PRO*), (74; RES*+PRO*), and (51; RES*+PRO*) have all have an average consumption of around 600 W, and an average standard deviation around 1,350 W. On the other hand, the Paris region has a lower consumption and standard deviation at 500 W (for the average consumption) and 1,000 W for the standard deviation.



(a) Average and median per hour of the day.



(b) Quantile Q25 hours of the day.

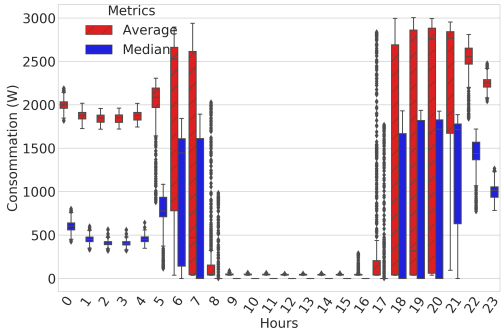


(c) Deviation standard per hour of the day.

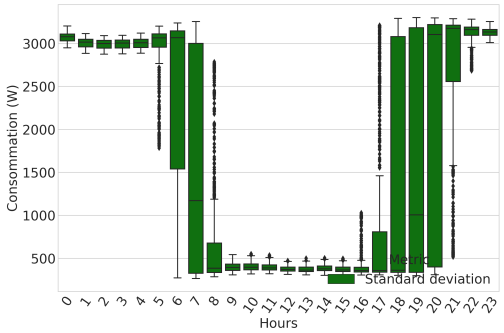
Figure 3.24 – *ENEDIS* dataset perimeter: (NAT; RESAutre)

The RESAutre case: The RESAutre meters (Figure 3.24) show quite atypical consumption happening mainly during the night, with two consumption peaks in the morning and the evening. The average consumption is around 1,000 W. This is especially visible for the quantile Q25, where the daily consumption is anecdotal and low at night (max 250 W). However, the standard deviation is not that much impacted by this day/night separation, with only the morning/evening peak that is visible.

The PRO5 case: PRO5 meters represent the public lighting meters. Therefore, as shown in Figure 3.25, it is normal to see them consuming only at night and not during the day. Their consumption is stable at night, with two peaks at the beginning and the end of the night. The transition period (the 4h-8h and 17h-21h periods) shows the transition between day and night due to the daytime seasonal variation. Standard deviation follows the meter consumptions closely. The unique distribution of the PRO5 consumptions explains the daily compression of the PRO* statistics.

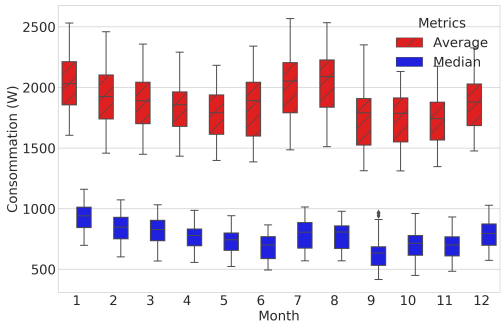


(a) Average and median per hours of the day.

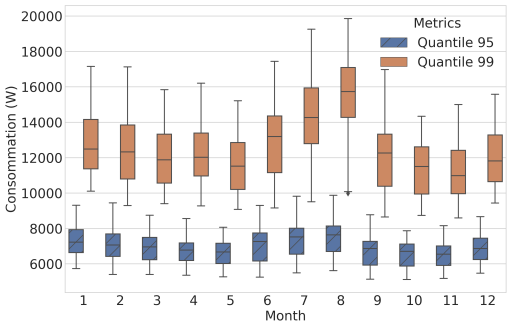


(b) Deviation standard per hours of the day.

Figure 3.25 – *ENEDIS* dataset perimeter: (NAT; PRO5)



(a) Average and median per calendar month.



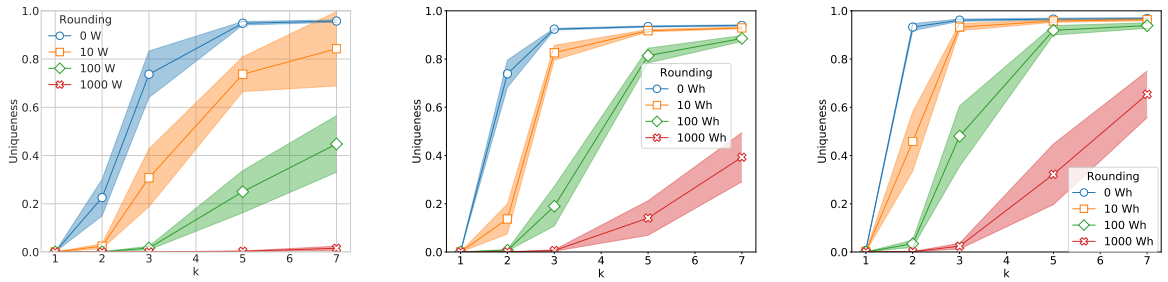
(b) Quantiles Q95 and Q99 per calendar month.

Figure 3.26 – *ENEDIS* dataset perimeter: (NAT; PROAutre) (1 = January, 12 = December).

The PROAutre case: The PROAutre meters distinguish themselves by having counter-seasonal consumption. Figure 3.26 shows a consumptions peak, evident for the quantiles Q95 and Q99, in the summer month and only a slight bounce during the winter.

3.5 Uniqueness

We begin this section by looking at the uniqueness of the whole datasets (*ENEDIS*, *ISSDA*, and *LONDON*). We obtain high uniqueness with a handful of timestamps for every dataset. Then, we look at the impact of the data degradation on the uniqueness. It shows the difficulties of significantly reducing uniqueness while preserving utility. Finally,



(a) (b) (c)

Figure 3.27 – Average uniqueness (mean with minimum/maximum uniqueness observed per timestamp) according to the number of consecutive points (k) and the rounding for (a) the half-hourly *ENEDIS* dataset, (b) the daily *ENEDIS* dataset and (c) the subset of the daily *ENEDIS* dataset restricted to time series generated by the smart meters having generated the half-hourly dataset.

we look at multiple parameters potentially explaining and impacting our results. We find that the dataset entropy is strongly correlated to uniqueness.

To illustrate why uniqueness reaches such high values in our datasets, we consider the three illustrative electricity consumption time series shown in Figure 1.2. Each series corresponds to a smart meter measuring power consumption in Watt (W) for a day with 48 measures (i.e., 30 minutes rate). The three series look similar because many individuals share everyday activities (e.g., waking up, commuting to work, and performing other activities roughly simultaneously). Their electricity consumption time series mirror these common behaviors. However, no household is the exact copy of another. First, the same activity of an electric device shifts along the time axis due to differences in household schedules. Also, using distinct electric devices results in distinctive electricity consumption. This contributes to making them *unique along the time*.

Figure 3.27 shows the average uniqueness on the *ENEDIS* datasets, according to k and the order of magnitude of the rounding. Note that in this section, the *ENEDIS* dataset contains only the RES* profiles. A very high uniqueness (above 90 %) is reached at $k = 5$ (i.e., 2h30 consumption) in the half-hourly dataset (Figure 3.27a), and 3 days - $k = 3$ - consumption in the daily dataset (Figure 3.27b and Figure 3.27c), without any rounding. It is worth noting that sub-sequences of length $k = 1$ have a very low uniqueness: almost every smart meter consumption is shared by at least another meter. However, as time passes, even by a few hours, almost every smart meter generates a different consumption time series, making them unique.

We further note that although the trends are similar, higher uniqueness can be observed on the (small scale) publicly available *ISSDA* and *LONDON* electricity consumption time series datasets. On the half-hourly *ISSDA* dataset, we reached 90 % uniqueness at $k = 3$. We reach this threshold on the daily *ISSDA* dataset with $k = 1$. On the half-hourly *LONDON* dataset, we reached 90 % uniqueness at $k = 3$. On the daily dataset, this threshold is also reached at $k = 3$, with more than 70 % uniqueness at $k = 1$. This can be explained by the small size of the datasets resulting in much sparser datasets, thus in fewer collisions, and as a result in higher uniqueness.

Focusing on Figure 3.27a, i.e., the uniqueness of the half-hourly *ENEDIS* dataset, we confirm the following intuition: the higher the rounding, the lower the uniqueness, which nevertheless remains far from being negligible even with the strongest degradation. For example, rounding to 2 orders of magnitude (i.e., to the closest 100 W) results in uniqueness above 40 % for $k = 7$. Rounding to the kilowatt (3 orders of magnitude), the uniqueness is about 0.5 %, corresponding to 12,500 unique time series. A non-negligible number of households. Rounding to the kilowatt is a very strong degradation given the average consumption and the standard deviation (see Section 3.4.1) Note that such a rounding results in a dramatic information loss. Although the definition domain of our electricity consumption measures is $[0;\ 36,000]\ W$, most measures fall between 500 W and $1,500\ W$, as illustrated in Section 3.4.1. Overall, the average consumption measure over the full year is 725 W with a mean, standard deviation of 950 W.

Focusing now on Figure 3.27b, i.e., the uniqueness of the daily dataset, we observe even higher uniqueness, whether rounding is enabled or not. Because rounding applies here to much higher measures in expectation (daily measures instead of half-hourly measures), it has (unsurprisingly) much less impact on uniqueness. After rounding to the kW the daily measures, around 40 % of the 25M meters are still unique with $k = 7$ (i.e., one week rounded daily consumption). Figure 3.27c shows the uniqueness of a subset of the daily dataset, including only the daily time series of the 2.5M smart meters involved in the half-hourly dataset. By comparing it to Figure 3.27b, we observe the impact on the uniqueness of the scale of the dataset. In general, thanks to its sparser space, the subset of the daily dataset reaches higher uniqueness. More precisely, increasing the number of time series by an order of magnitude (i.e., from 2.5M to 25M) slightly reduces uniqueness: obtaining 90 % uniqueness requires $k = 3$ consecutive measures without rounding (instead of $k = 2$), requires $k = 5$ consecutive measures with rounding to the 10 Wh (instead of $k = 3$), and requires $k = 7$ with rounding to the 100 Wh (instead of $k = 5$). With the

highest rounding enabled (to the kWh) and when $k = 7$, uniqueness drops from around 70 % on the daily subset to 40 % on the full daily dataset.

Note that the minimum and maximum uniqueness may vary more or less depending on k and the order of magnitude of rounding. For example, for the half-hourly dataset, with $k = 3$ and no rounding, the minimum and maximum uniqueness differ by 40 % at most, while for most parameter settings, the difference between the minimum and maximum uniqueness remains small.

Finally, we observe that the uniqueness reaches a maximum value (e.g., around 96 % for both the half-hourly and daily datasets). Once this value is reached, increasing k only increases the uniqueness by a tiny percentage. This can be explained by the significant proportion of electricity consumption measures equal to 0 W (between 3 % and 10 %).

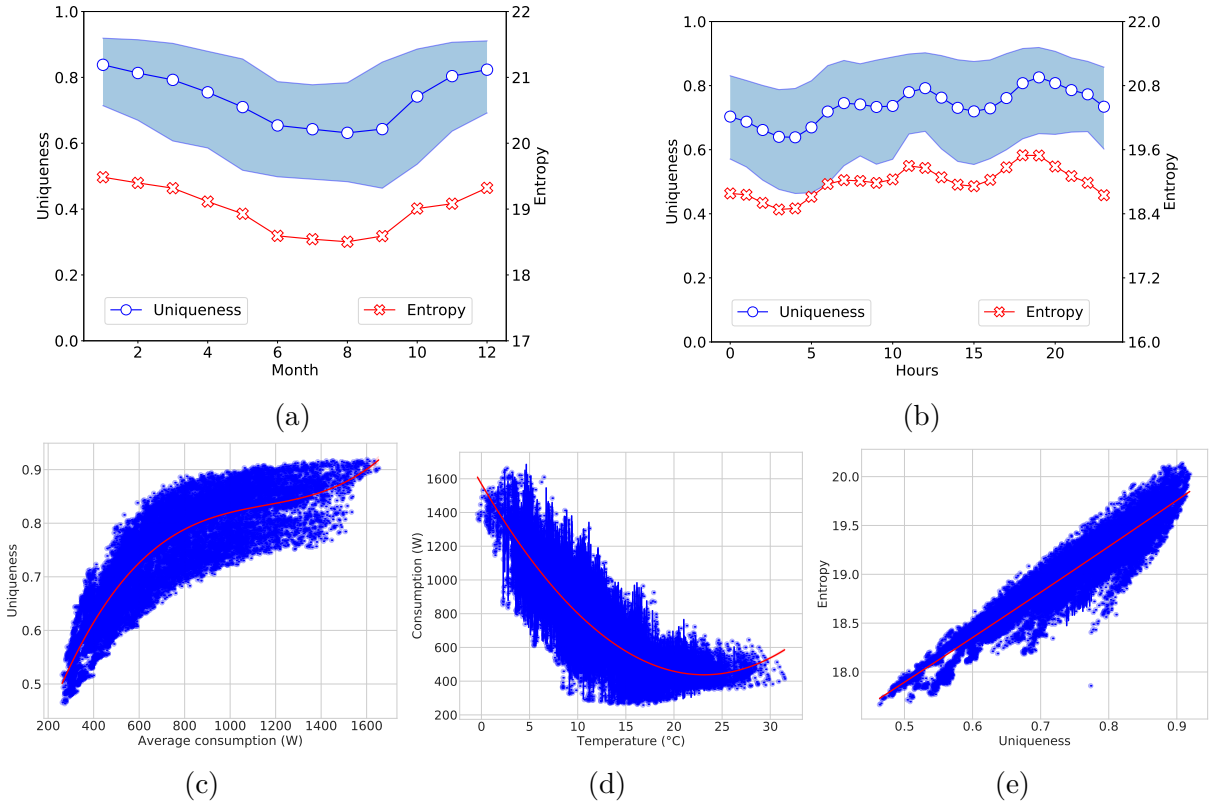


Figure 3.28 – Monthly (a) and hourly (b) uniqueness along time (mean with the minimum / maximum uniqueness observed), together with the relationship between electricity consumption and uniqueness (c), between temperatures and electricity consumption (d), and between entropy and uniqueness (e). All figures are computed based on the half-hourly *ENEDIS* dataset with $k = 3$ and no rounding (i.e., to the Watt (W)).

Figure 3.28 shows how uniqueness varies with time. In particular, Figure 3.28a shows a first seasonal variation leading to higher uniqueness in winter than in summer. Figure 3.28b shows that uniqueness is higher during the day than during the night with two peaks, the first peak at noon (12h) and the second peak in the evening (18h-22h). The entropy of the sets of k-length subsequences at each timestamp is also plotted on Figures 3.28a and 3.28b in order to observe visually the correlation between uniqueness and entropy. This is also clearly illustrated in Figure 3.28e. The Pearson correlation coefficient between uniqueness and entropy is 0.94 (the closer to 1 the stronger the correlation). In addition, as shown in Figures 3.28c and 3.28d, uniqueness is also strongly correlated to the average consumption (Pearson coefficient of 0.8) which is itself strongly (negatively) correlated to the temperatures (Pearson coefficient of -0.8): the lower the temperature the higher the consumption, and the higher the consumption the higher the uniqueness. The consumption seems to stabilize for an average temperature above 15 °C. At local levels, a temperature above 30 °C leads to a consumption bounce. However, France is not warm enough to be able to spot such patterns on a nationwide scale. In a nutshell, uniqueness is higher for a winter evening than for a summer night.

3.6 Discussion

This study aimed at presenting statistical metrics about the three electricity consumption datasets used in this document (i.e., *ENEDIS*, *LONDON*, and *ENEDIS*). The first two are relatively small (around 5,000 series) but, contrary to the *ENEDIS* datasets, they are public. Therefore, we use them as reproducible caution for the works described in this document. The large-scale *ENEDIS* dataset is the most valuable as it contains millions of real-life consumption measurements.

The analysis of the datasets shows that most of the studied consumptions follow similar patterns. The individual consumption and its variance per POM (measure points) are strongly correlated for most perimeters. Specifically, when the average consumption increases, the consumptions are more dispersed. The analysis highlighted two seasonal patterns. First, an intra-day pattern where the consumption mostly takes place during the day instead of at night with a consumption peak in the evening (except for the *LONDON* and *ENEDIS PRO5* perimeters). Then, a seasonal pattern correlated with the temperature: the colder it is, the higher the consumption. Again, the *LONDON* datasets stand out by not being sensitive to the temperature. Surprisingly, there is no notable

difference between weekdays on the global level but slight differences between weekdays and weekend days over some *ENEDIS* profiles.

The study presents the first uniqueness results computed over two large-scale, nationwide electric consumption time series datasets. We show that uniqueness reaches high values even when computed over tiny subsequences. Our results hold despite the following adverse facts. First, individuals behave similarly (e.g., sleeping patterns and commuting periods). While this (relative) uniformity can be observed in the results that only include single consumption measures ($k = 1$), uniqueness increases fast when considering only a little bit more consumption measures ($k > 1$). Second, individual measures are concentrated over a small part of the definition domain. In the event of the deployment of new and more precise meters, we expect uniqueness to rise as the definition domain widens and consumption measurements become sparser. Again, this impacts our results when uniqueness is computed on single electric consumption measures, but considering only a few additional measures makes uniqueness rise tremendously. Third, we compute uniqueness at each timestamp t based on k-length subsequences. Performing an exhaustive search of the subsets of k measures, not necessarily consecutive, that leads to the highest uniqueness might indeed lead to higher uniqueness. However, we believe that the uniqueness results that we obtain are already sufficiently high for raising strong concerns about the re-identifiability of households within electric consumption time series datasets. Fourth, to the best of our knowledge, our datasets are the largest electric consumption time series datasets (i.e., around 2.5M half-hourly time series and around 25M daily time series – both during one year). Despite the possible collisions that increase with the number of time series in the dataset, uniqueness remains dramatically high, even when considering small subsequences.

Our study also shows the impact on uniqueness of degrading the time series severely. Surprisingly, uniqueness remains high despite losing orders or magnitude precision and considering small subsequences. Even when uniqueness drops (i.e., rounding the half-hourly series to the kW), around $12,500$ thousand of households among the 2.5M of the full half-hourly dataset remain unique. This shows the limitation of naive protection methods (e.g., rounding) from potential re-identification attacks.

The high uniqueness depicted in the results of this study shows that re-identification of households in large-scale electric consumption datasets might be possible with high probability by adversaries knowing only a small subset of consumption measures of their target(s). As seen in Chapter 1, detailed electricity consumption measures can be cap-

tured today by a wide range of actors besides the grid manager. Although all these actors may not adopt adversarial behaviors, the attack surface is large, with weaknesses, and they may suffer from negligence. This increases the risk of leaking electric consumption measures to the wild. Additionally, even rough estimates of electric consumption measures might be sufficient for performing a re-identification (e.g., based on other sources of information about the household, based on past data, based on a subset of the electric consumption). Adversaries with approximate knowledge might still benefit from the uniqueness of degraded electric consumption time series for performing re-identifications.

Our uniqueness results are based on two French large-scale electric consumption datasets and two public electricity consumption datasets. While we know biases might impact the actual uniqueness numbers (e.g., climate biases, socio-cultural biases, political biases), we believe similar conclusions about uniqueness can be drawn from many other large-scale electric consumption datasets. Indeed, the climate is the main driving force for households' energy consumption and, consequently, for uniqueness. However, our results show that for $k \geq 5$, worryingly high uniqueness levels are reached (above 90 %), independently from time and climate.

Chapter 4

The SubSum attack

This chapter introduces our *SubSum* attack, a membership inference attack. This attack uses a threat model where the attacker has access to the published aggregates (e.g., averages, sums) associated with additional attributes and a set of disclosed time series. This attack can be formalized as an extension of the subset-sum problem [KPP04]. Specifically, each time point is used as a constraint to find the series at the origin of an aggregate. We use constraint programming to solve the problem. Based on this approach, we perform experiments on the *ISSDA* and *LONDON* datasets (see Chapter 3) using the Gurobi solver. We show that our membership inference attack is usually successful. Our findings demonstrate that breaking privacy when aggregate-based techniques are used to publish time series is much easier than expected. We analyze the limit cases to help publishers enforce individuals' privacy. This attack was published at the *Conference of Security and Cryptography (SECRYPT)* [Voy+22a].

This chapter is organized as follows: Section 4.1 contains the background knowledge. Section 4.2 presents the problem and defines the threat model. Section 4.3 presents the attack algorithm. Section 4.4 contains a large-scale experimental analysis of the attack. Section 4.5 concludes and presents interesting future works of this attack.

4.1 Background knowledge

Constraint programming (CP) [Apt03] is a programming technique that can be used to model and solve mathematical optimization problems. The problem is modeled by a set of variables and equations called constraints. A **variable** is an unknown value within a definition domain (i.e., a set of numbers). A **constraint** is a logical relation between the variables. Constraints restrict the possible values a variable can take. The model's objective can be to prove (or disprove) the feasibility of the problem, or to find real solutions, if any. A **solution** is a set of variable values matching the constraints.

Mixed Integer Programming (MIP) [Wol20] focuses on problems where all the variables are integers. MIP and CP work in different ways. Both uses a branch-and-bound search procedure that explores the space of feasible solutions and iteratively refines the solution until an optimal solution is found. MIP uses algebraic techniques (such as linear relaxation), while CP uses logical inferences. It is generally accepted that MIP solvers are faster at finding real solutions on linear problems while CP is more flexible and can handle a wider range of constraints. Multiple solvers can solve MIP problems. Gurobi [1] is recognized as the most performant solver [Jab+15]. We note the existance of three main competitors: OR-Tools [2] and CPLEX [3] and Choco [4].

The **Subset-sum problem** [KPP04] is a combinatorial optimization problem. It is a well-known NP-hard problem. The objective is to determine whether a subset from a list of numbers (W) can sum to a targeted value (o). For example, given bag able to carry 10 kg ($O = 10$) and the following set of numbers $W = \{5, 2, 1, 3, 6\}$, the following set of numbers (i.e., solutions) can fill the bag: $\{5, 3, 2\}$ and $\{6, 3, 1\}$.

The subset-sum problem can be modelized and efficiently solved as a MIP problem [Wol20]. Equation 4.1 represents the sumset-sum problem as MIP, with W the set of weighted objects and $V_i \in \{0; 1\}$ the variable indicating if the object i has been selected.

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in |W|} V_i \cdot W_i \\ \text{subject to} \quad & \sum_{i \in |W|} V_i \cdot W_i = o \end{aligned} \tag{4.1}$$

4.2 Problem statement

4.2.1 The publishing environment

Publisher: The publishing environment consists of a **publisher** and an **attacker** (described in Section 4.2.2). Figure 4.1 illustrates the publishing environment. The publisher collects from a set of **individuals** $\mathcal{I}$ a set of time series $\mathcal{S}$ (see Chapter 3). Each individual produces a single fixed-length time series. Each time series is associated with, possibly

1. `http://www.gurobi.com`
2. `https://developers.google.com/optimization/`
3. `https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer`
4. `http://www.choco-solver.org`

Notation	Description
$\mathcal{I}$	Set of individuals
$\mathcal{T}$	Set of timestamps
$\mathcal{S}$	Set of time series, $\mathcal{S}_{i,t}$ is the measurement of the individual i at timestamp t.
$\mathcal{S}^{\mathcal{A}} \subset \mathcal{S}$	Set of time series participating in the aggregates.
$\mathcal{A}$	Vecotor of aggregates, $\mathcal{A}_t$ the aggregate at timestamp t.
θ	Attacker's time budget
p	Pool size (number of solutions asked to the solver)
$\mathcal{P}$	Set of solutions found by the attacker
$\mathcal{G}$	Adversarial guesses

Table 4.1 – Notations

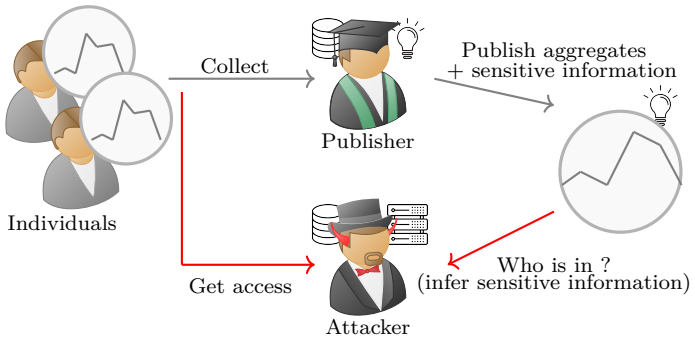


Figure 4.1 – The *SubSum* Attack

privacy-sensitive, additional attributes contextualizing the time series (e.g., income, house surface, socio-demographic information, electricity appliances).

Published aggregates: The publisher selects a subset $\mathcal{S}^{\mathcal{A}} \subset \mathcal{S}$ and computes and publishes one aggregate $\mathcal{A}_t$ per timestamp over $\mathcal{S}^{\mathcal{A}}$. In this chapter, we focus on aggregates that are *sums* (we use the two terms indistinctly below). Note that all sum-based aggregation methods (e.g., average) can also be tackled by our attack with minor changes in the algorithm. Formulae 4.2 computes the aggregates for each timestamp t by summing the values of each individual in $\mathcal{S}^{\mathcal{A}}$.

$$\mathcal{A}_t = \sum_{\forall i} \mathcal{S}^{\mathcal{A}}_{i,t} \tag{4.2}$$

The set of individuals selected in $\mathcal{S}^{\mathcal{A}}$ represents the **ground truth**. We model the ground truth as a vector of probability. This vector tells which individuals of $\mathcal{S}$ are part

of $\mathcal{S}^{\mathcal{A}}$. The probability of the i^{th} individual is set to 1 if and only if the i^{th} individual is in $\mathcal{S}^{\mathcal{A}}$, and to 0 otherwise. It is worth noting that the aggregates $\mathcal{A}$ usually come with the timestamps and the amount of aggregated time series.

Published additional attributes: To make the published aggregates more valuable, they are usually associated with additional information that characterizes the subpopulation in the aggregates. Such characterizing attributes are described in Chapter 1 and Section 2.1.

4.2.2 Threat model

The usual threat model considers an attacker who aims to retrieve the identity (or associated valuable information) of the individuals concerned by the publication [Jay+21; BB20; PTC18]. We consider a threat model where the attacker has access both to the (published) aggregated time series and to the (leaked) individual time series (while the additional attributes remain secret and the goal of the attack remains unchanged). We now discuss two points: the relevance of this scenario and the attacker's motivation to perform an attack in such a scenario.

Due to legal requirements and widespread security best practices, publishers usually store direct identifiers and individual time series in separate databases applying, for example, traditional pseudonymization schemes (see Section 3.5). Dedicated access control policies, archival rules, or security measures can be applied to each database (e.g., a retention period limited to a couple of years for fine-grained electricity consumption time series). Isolating direct identifiers from time series prevents direct re-identifications from malicious employees or external attackers exploiting a leak. The same security principles apply to storing additional, possibly sensitive, attributes. Note that our model can easily capture variants of the above context. An example of a variant, common in real-life, is when the additional attributes come from another data provider collaborating with the publisher for computing cross-database statistics. Whatever the scenario, this results in a set of databases isolated from each other for security reasons to mitigate the impacts of data leaks or simply because multiple data providers collaborate. Data leaks are common, either due to insider attackers or external ones. Considering that raw data may eventually leak is a conservative approach that should be explored. If the identifiers, the time series, and the additional attributes all leak together, attacking the aggregates is useless. However, in cases where only the raw times series leak, attackers need to deploy membership

inference attacks to learn the association between a target individual and its attributes. In the following, we consider this threat model. We show in a real-life example that an attacker can efficiently retrieve the attributes of a target using linear programming.

Formally, the adversarial background knowledge consists of the complete set of time series $\mathcal{S}$. Given his/her background knowledge $\mathcal{S}$, the vector $\mathcal{A}$ of timestamped aggregates, and the amount of time series aggregated, the attacker outputs a vector of **guesses** $\mathcal{G}$ over the participation of each individual to the vector of aggregates. The vector of guesses is similar to the ground truth vector s.t. the i^{th} value of the vector of guesses represents the probability of the participation of the i^{th} individual to the aggregate. The closer the guesses are to the ground truth, the more successful the attacker is.

4.3 The *SubSum* attack

The *SubSum* attack is based on finding solutions to the subset-sum problem [KPP04] over the aggregates $\mathcal{A}$ given the set of time series $\mathcal{S}$. An overview of the problem is available in Section 4.1.

The *SubSum* attack finds solutions to the subset-sum problem. The subsets-sum problem is NP-hard. Solving this problem is inefficient and would not scale in the general case. This difficulty is because an important set of individuals may have similar values for a given timestamp, which may further increase combinatorics. By exploiting the different time points (with different aggregate values but originating from the same set of individuals), the solver can prune the unsatisfactory solutions much faster and converge, with enough constraints, to possibly unique solutions if all individuals do not have identical values as another individual in all-time points.

Algorithm 1 thus combines multiple subset-sum problems, one for each aggregate (each timestamp), into a single set of IP constraints. Let X be a vector of $|\mathcal{S}|$ Boolean variables denoting the time series that are (possibly in case of multiple solutions) part of the aggregate. We define one constraint per aggregate $\mathcal{A}_t$ in Equation 4.3.

$$\sum_{\forall i \in \mathcal{S}} (X_i \cdot \mathcal{S}_{i,t}) = \mathcal{A}_t \tag{4.3}$$

We define one final constraint representing the attacker's knowledge of the number $|\mathcal{S}^{\mathcal{A}}|$ of individuals present in the aggregates in Equation 4.4.

Algorithm 1: *SubSum* attack

Input: The set of time series ($\mathcal{S}$), the aggregates ($\mathcal{A}$), the time budget (θ), the number of solutions to look for (p)

Output: The solutions found ($\mathcal{P}$), the guesses ($\mathcal{G}$), the status code (`status`)

set_time_limit(θ*)*
set_pool_size(p*)*
for $i \in |\mathcal{S}|$ **do**
  add_variable(X_i*,* $\{0, 1\}$*)*
end
for $t \in |\mathcal{A}|$ **do**
  add_constraint($\sum_{\forall i \in \mathcal{S}} X_i \mathcal{S}_{i,t} = A_t$*)*
end
add_constraint($\sum_{\forall i \in \mathcal{S}} X[i] = |\mathcal{S}^{\mathcal{A}}|$*)*
$\mathcal{P}$:= *solve()*
$\mathcal{G}$:= *to_probability_vector(*$\mathcal{P}$*)*
status := *get_status()*
return $\mathcal{P}$, $\mathcal{G}$, *status*

Functions used in the algorithm :

***add_variable(bounds)*:** Define an integer variable bounded to a given set of values.

***add_constraint(constraint)*:** Define a constraint binding the model.

***set_pool_size(*p*)*:** Define an upper bound p on the number of solutions to search for.

***set_time_limit(*θ*)*:** Set the time budget θ of the solver.

***solve()*:** Solves the problem and get the solutions found by the solver (if any).

***get_status()*:** Get the status of the solver after its termination. We assume that at least the two following statuses are available : *OPTIMAL* (i.e., either p solutions are found or less than p solutions are found but no more solutions exist), *TIMELIMIT* (i.e., the time budget θ is over) and *INFEASIBLE* (i.e., the model cannot be solved).

$$\sum_{\forall i \in \mathcal{S}} X[i] = |\mathcal{S}^{\mathcal{A}}| \tag{4.4}$$

Our objective is to find less than p solutions (ideally $p = 2$) to infer meaningful information about the aggregate members within a reasonable time (i.e., before reaching the wall time θ). The solver runs until one of the following conditions is satisfied: (1)

all the existing solutions are found, (2) the maximum number of solutions p is reached, and (3) the time budget θ is exhausted. The solver outputs a set $\mathcal{P}$ of up to p candidate solutions together with a status code. Each solution is a probability vector, similar to the ground truth vector, indicating the individuals from $\mathcal{S}$ that, when summed up on the proper timestamp(s), solve the given set of constraints X. Finally, the attacker computes the frequency of each individual in the set of solutions to obtain his guesses $\mathcal{G}$.

4.4 Experiments

We perform an extensive experimental study of the attack using two open datasets: the *ISSDA* dataset (see Section 3.2) and the *LONDON* one (see Section 3.3). Our results show that the *SubSum* attack finds the exact set of individuals originating the aggregate when the attack's requirements are met. The attack requires the published aggregates to contain half the number of timestamps as the number of series known to the attacker.

4.4.1 Experimental settings

Experimental environment: The code of our experiments is publicly available [5]. It consists of two software modules. First, the *driver* module, written in Python, is in charge of driving the complete set of experiments: deploying the experiments, launching them, stopping them, and finally fetching the experimental results. We deploy the *driver* module on our own OAR2 (Linux) [6] computing cluster, allocating at least 2 cores (2.6 GHz) and 2 GB RAM to each experiment. Each experiment is a *SubSum* attack for a set of parameters. We describe it below. The second software module is the solver performing a *SubSum* attack given a set of parameters. We use in our experiments the Gurobi solver [7], a well-known and efficient IP solver, but any other solver can be used provided that it implements an API similar to the one used in Algorithm 1. We perform our experiments over the real-life *ISSDA* and *LONDON* public electricity consumption datasets (defined in Section 3.2 and Section 3.3 respectively).

Experimental protocol: We consider three main parameters impacting the *SubSum* attack: the size of the set of time series ($|\mathcal{S}|$), the length of the aggregates published

5. `https://gitlab.com/phd_antonin/subsum`
6. `https://oar.imag.fr/`
7. `http://www.gurobi.com`

Parameter	Value
$\mathcal{S}$	$\{ISSDA, LONDON\}$
$\lvert\mathcal{S}\rvert$	{1000, 2000, 3000, 4000, 4500}
$\lvert\mathcal{S}^{\mathcal{A}}\rvert$	$\{0\ \%, 10\%, \ldots, 100\ \%\} \times \lvert\mathcal{S}\rvert$
$\lvert\mathcal{T}\rvert$	$\{0\ \%, 10\ \%, \ldots, 100\ \%\} \times \lvert\mathcal{S}\rvert$
θ	{1000, 2000, 4000, 8000, 86400}
p	{2, 100}

Table 4.2 – Values of the parameters used in our experiments.

(number of constraints/timestamps $|\mathcal{T}|$), and the number of series in the aggregate ($|\mathcal{S}^{\mathcal{A}}|$). Table 4.2 shows the parameter values we use in our experiments. For each triple of parameters (i.e., $|\mathcal{S}|$, $|\mathcal{T}|$, $|\mathcal{S}^{\mathcal{A}}|$), we generate 20 experiments [8] where each experiment is a *SubSum* attack over (1) a set of time series, (2) a vector $\mathcal{A}$ of aggregates published, and (3) a time budget θ. At each experiment, the whole population $\mathcal{S}$, together with the individuals $\mathcal{S}^{\mathcal{A}}$ that are part of the aggregates, are randomly selected. We log for each experiment the ground truth (i.e., the exact set of individuals in $\mathcal{S}^{\mathcal{A}}$), the candidate solution(s) found by the solver, the status of the solver, and the wall-clock time elapsed during the attack.

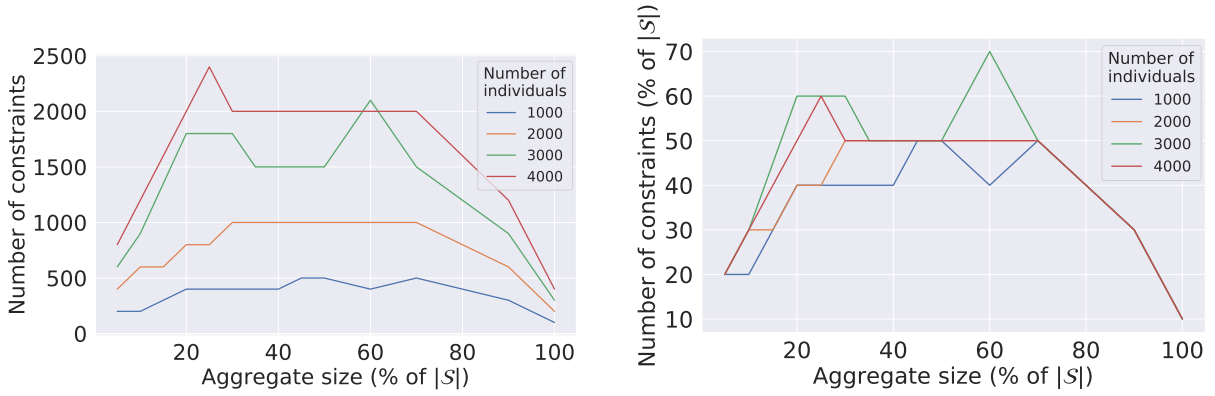
Success definition: Our experimental validation considers a flexible success definition. An attack is considered to be a **success** if the solver finds strictly less than p solutions in the allowed time θ [9]. Indeed, this implies (1) that the solution is part of the pool of solutions and (2) that the solver found all the solutions. We mainly focus on a pool size equal to 2. In this case, success occurs when the solver outputs a single solution: the membership inference is perfect. However, a pool size equal to 2 is not always sufficient. In these cases, we increase the pool size and analyze the solutions more deeply found by the solver, showing the distribution of the guess vector on the individuals that are part of the ground truth.

8. Performing 20 times the *SubSum* attack on the same set of parameters is sufficient for obtaining a stable success rate for the given set of parameters.

9. Note that this is a somewhat precautionary success measurement since the attacker concludes when he/she is sure the solver has found all solutions. Other success metrics can be considered in which the attacker gains information from incomplete sets of solutions the solver returns. However, when the solver reaches the time budget or when the pool p is filled, the solver fails to prove the non-existence of other solutions. New yet undiscovered solutions might refute inferences on incomplete results.

4.4.2 Experimental results

Our experiments aim at providing clear insights on the conditions leading to successful *SubSum* attacks. First, we study the *SubSum* success rate according to the total number of time series in the entire population (population size $|\mathcal{S}|$), the number of time series aggregated in the published aggregates (aggregate size $|\mathcal{S}^{\mathcal{A}}|$), and the length of the series (number of constraints $|\mathcal{T}|$). We set the time budget to a value sufficiently high to not interfere with the attack. This study shows a clear relationship between the population size and the published aggregates (size and length) for a successful *SubSum* attack. This relationship allows us to express both the size of the published aggregates and its length relatively as a fraction of the population size. Then, we study the impact of the time budget (θ) on the success rate. Finally, we study the efficiency of the *SubSum* attack on the entire population available in our datasets.



(a) Y axis: absolute number of constraints

(b) Y axis: number of constraints as a percentage of the number of individuals

Figure 4.2 – Minimal number of constraints before getting at least one success. Dataset = *ISSDA*, $|\mathcal{S}| = \{1000, 2000, 3000, 4000\}$, θ=24h, p=2, 20 repetitions.

Influence of $|\mathcal{S}|$ on the success rate: Figure 4.2 shows the minimal constraints required for at least one success for several population sizes and for a time budget large enough to be neutral (i.e., $\theta = 24h$). Even if the required number of constraints before achieving success increases with the population size, it follows a parabolic shape for all population sizes. Figure 4.3 shows the success rate for the aggregate size and the number of constraints for a small dataset of 1000 individuals and a small time budget of 1000s (approximately 15 minutes).

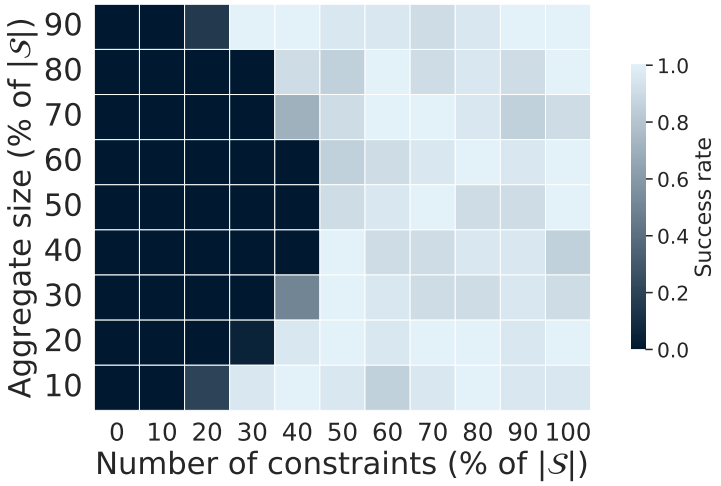
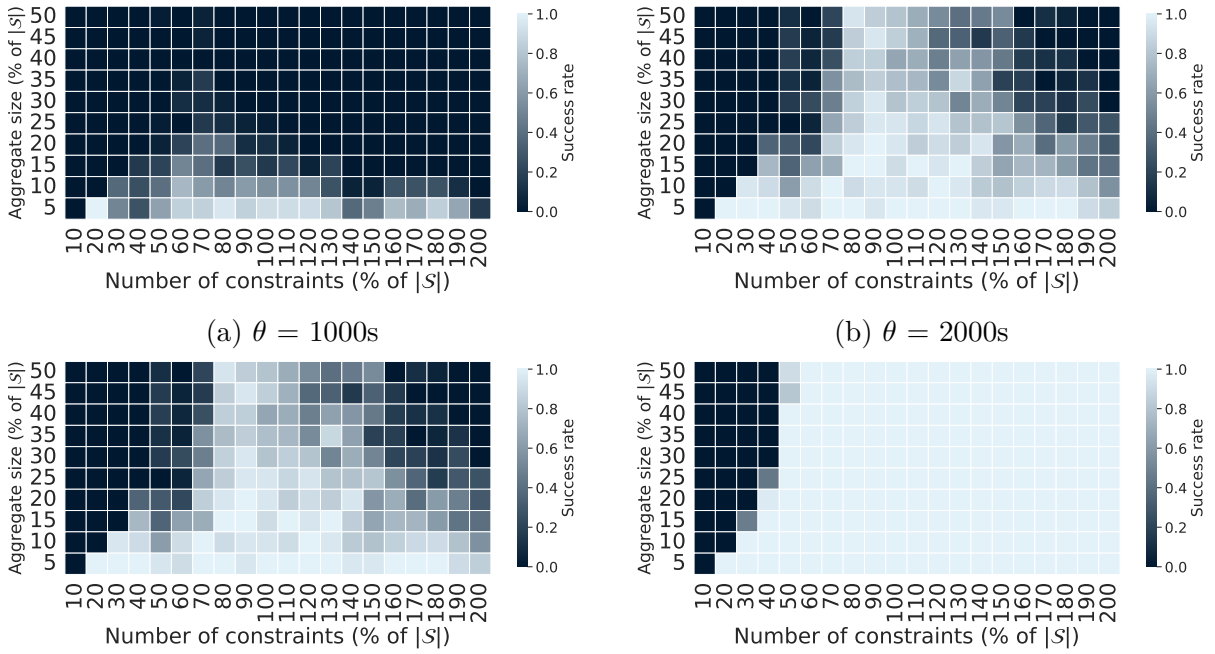


Figure 4.3 – Success rate (0 : all failed, 1 : all succeed). Dataset = *ISSDA*, $|\mathcal{S}| = 1000$, θ=1000s, p=2, 20 repetitions.

As already seen in Figure 4.2, Figure 4.3 shows that the attacks fail in a parabolic area centered around aggregates of half the population size because the solver reaches the maximum allocated time. Outside this area of failure, the attacks are mostly successful except for a few randomly distributed, failed experiments that depend on the original aggregate and population samples. In this "light" area, the few failures are due to the existence of several possible solutions (we provide below an advanced analysis of the cases where several solutions exist). After reaching an aggregate size corresponding to 50% of the population, the number of constraints needed is expected to decrease again. Identifying $|\mathcal{S}^{\mathcal{A}}|$ individuals in $\mathcal{S}$ is as difficult as identifying $|\mathcal{S}| - |\mathcal{S}^{\mathcal{A}}|$ individuals. With small aggregate sizes (lower than 50% of the population size), the number of constraints needed is often twice the size of the aggregate: in general, the number of constraints needed to successfully attack a population of size $|\mathcal{S}|$ should be *a*t least in the order of the aggregate size $|\mathcal{S}^{\mathcal{A}}|$. Conversely, if the number of constraints is much lower than the aggregate size, the dataset is deemed safe (relatively to the time budget θ and to the pool size p) to the *SubSum* attack. Building on this result, in the following, both the aggregate size ($|\mathcal{S}^{\mathcal{A}}|$) and the length of the series (number of constraints $\mathcal{T}$) are expressed as a fraction of the population size ($|\mathcal{S}|$). Note that the number of published timestamps might be larger than the population size, resulting in fractions larger than 100% of the population size.



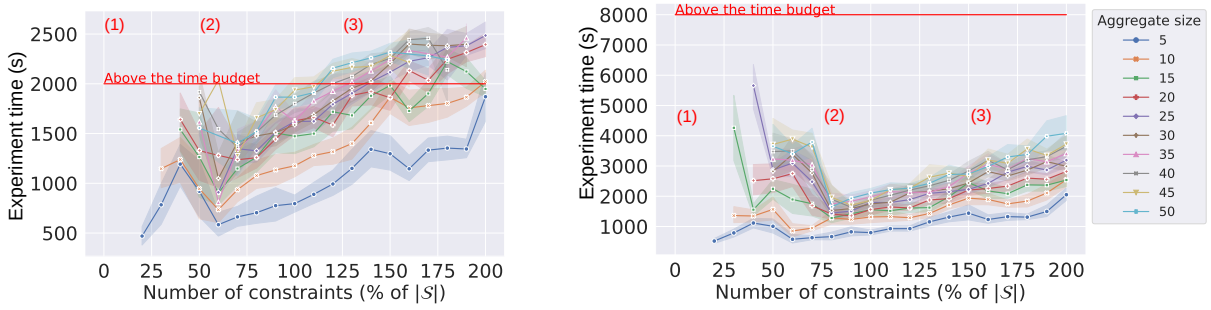
(a) $\theta = 1000$s

(b) $\theta = 2000$s

(c) $\theta = 4000$s

(d) $\theta = 8000$s

Figure 4.4 – Evolution of the success rate (0: all failed, 1: all succeed) depending on the time budget. Dataset = *ISSDA*, $|\mathcal{S}| = 2000$, $\theta = \{1000\text{s}, 2000\text{s}, 4000\text{s}, 8000\text{s}\}$, $p = 2$, 20 repetitions.



(a) $\theta = 2000$s

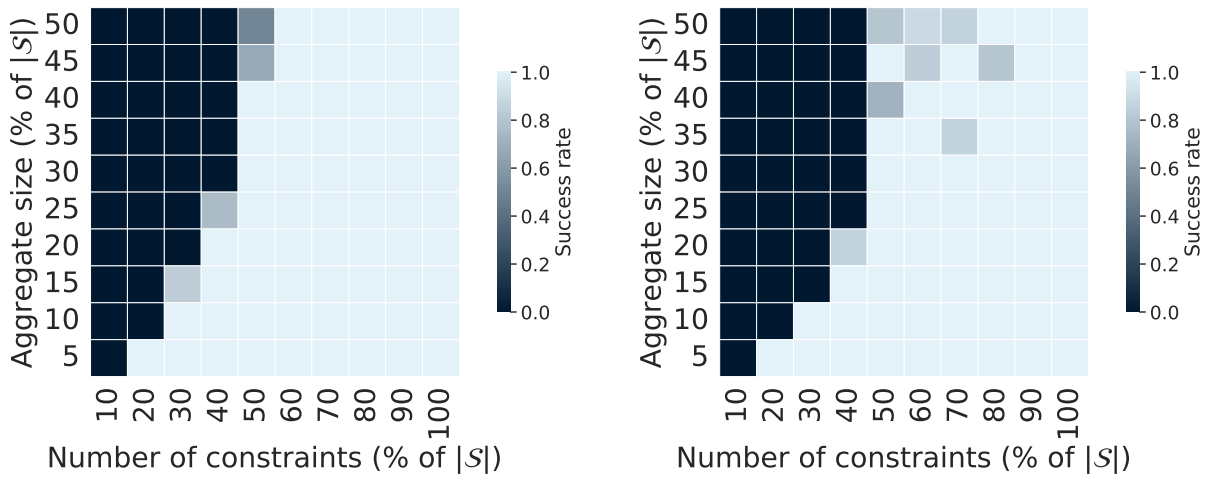
(b) $\theta = 8000$s

Figure 4.5 – Evolution of experiments time depending on the time budget. Dataset = *ISSDA*, $|\mathcal{S}| = 2000$, $\theta = \{2000\text{s}, 8000\text{s}\}$, $p = 2$, 20 repetitions.

Impact of the time budget θ: For practical reasons, we evaluate the impact of the time budget on a relatively small dataset with $|\mathcal{S}| = 2000$. As seen in Figure 4.4, the time budget θ impacts the attack success: the higher the time budget, the higher the success rate. However, as shown in Figure 4.4d, even with a high time budget, attacking an aggregate with a few time points (the left dark sides of Figures 4.4b, 4.4c and 4.4d) is unsuccessful. Conversely, if the time budget is too low and the number of constraints is too high (the right black/grey area of the Figures 4.4a, 4.4b and 4.4c), the solver does not have the time to solve the problem with its given parameters. Figure 4.5 shows the experiment's time (in seconds). We note three areas of interest on the figures: an area with missing points (before reaching 25% of constraints) corresponding to the previously defined, parabolic area of failures (Figure 4.2 and Figure 4.3) (1). The failures are due to reaching the solver's time budget. When increasing the number of constraints, the execution time drops to a minimum for all aggregate sizes (between 25% and 100% of constraints) (2). After this point, the execution time increases linearly with the number of constraints until the execution time reaches the time budget again (Figure 4.5b) (3). There is an apparent compromise to make between the number of constraints used (that needs to be in the order of the aggregate size) and the time budget. If the attacker has a short time budget, he/she might prefer not to use all the available constraints at his/her disposal.

Attacking large populations: Previously, we attacked small subsets of the *ISSDA* and *LONDON* datasets. The attacks on the complete datasets provide results similar to the ones previously described. However, since the datasets are larger, we need to increase the time budget to 24h to observe the same phenomena. As shown in Figure 4.6, the relationship between the dataset size, the aggregate size, and the number of aggregates required for the attack to succeed is similar to what we observed in the other experiments. In particular, in this case, we need roughly twice more constraints than the aggregate size until we reach an aggregate size of 25% of $|\mathcal{S}|$. After that, the number of constraints needed to obtain a success remains stable at 50% of $|\mathcal{S}|$.

Advanced success analysis: We set the pool size to $p = 100$ in these experiments, meaning that the attack is considered successful if the number of solutions found is between 1 and 99. This allows us to study the cases where several solutions are found, making $\mathcal{G}$ worth analyzing. We chose the experiment in which the solver returns the high-



(a) Dataset = *LONDON*

(b) Dataset = *ISSDA*

Figure 4.6 – Attack success rate (0: all fail, 1: all succeed) for datasets of 4500 individuals. Dataset = {*ISSDA*, *LONDON*}, $|\mathcal{S}| = 4500$, $\theta = 24$h, $p = 100$, 20 repetitions.

est number of solutions and where the intersection of these solutions is the smallest. In other words: where the number of individuals that are part of several solutions is the highest. This case occurs in one of the 20 repetitions of the attack with the following parameters: dataset = *ISSDA*, $|\mathcal{S}| = 4500$, $\theta = 24$h, $p = 100$, $|\mathcal{S}^{\mathcal{A}}| = 225$ (5%) and $|\mathcal{T}| = 900$ (20%). It results in a pool containing 3 solutions. In Figure 4.7, we represent the part of the guess vector limited to the individuals present in the 3 solutions. Figure 4.7 is a histogram showing the fraction of individuals (Y-axis) associated with eleven possible ranges of guess in $\mathcal{G}$ [10] (X-axis). The figure shows that 224 (98.7%) individuals appear in all the solutions: their membership can thus be completely inferred with certainty. Only one individual differs in all three solutions: the membership probability of this individual is thus equal to 1/3. This shows that the *SubSum* attack gains significant knowledge about the aggregate members even when multiple solutions are provided.

4.5 Conclusion

Membership inference attacks aim to infer whether a specific individual belongs to a given aggregate. This work introduces a technique to perform membership inference attacks on (threshold-based) aggregated time series. Given a publicly known dataset of

10. Recall that the vector of guesses $\mathcal{G}$ contains the probability that the individual participates to $\mathcal{A}$ for each individual.

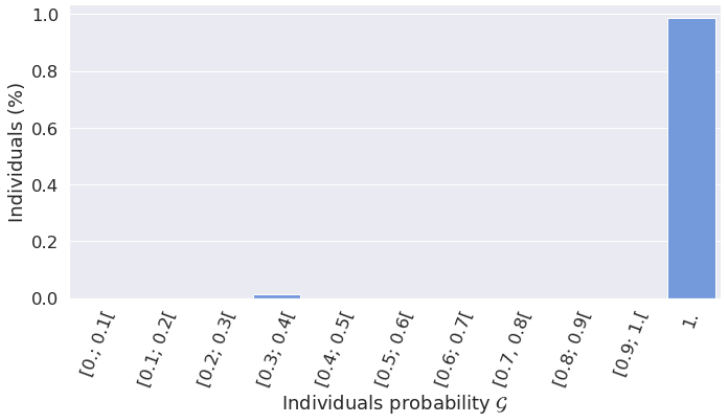


Figure 4.7 – Focus on the experiment in which the solver returned the highest number of solutions and where the intersection of these solutions is the smallest (3 solutions found in this case). We show the fraction of individuals associated with each possible adversarial guess value (organized within eleven ranges). Dataset = *ISSDA*, $|\mathcal{S}| = 4500$, θ = 24h, $p = 100$, $|\mathcal{S}^{\mathcal{A}}| = 225$ (5%), $|\mathcal{T}| = 900$ (20%).

time series and aggregated values (one value per timestamp) calculated from a private subset of that dataset, the adversary can re-identify the individuals belonging to the private subset. To do so, we model the membership inference attack as a subset-sum problem and use Gurobi to solve it. We perform experiments on two real-life datasets containing power consumption time series from UK and Ireland, respectively. We show that if the number of published timestamps is larger than the aggregate size (i.e., the number of individuals in the private subset), then the *SubSum* attack is highly likely successful.

Chapter 5

The STATS attack

In this chapter, we present the **S**hadow **T**raining for **A**ggregated **T**ime **S**eries (*STATS*) attack aiming to predict whether a target individual belongs to an aggregate or not. Unlike other membership inference attacks (MIA, presented in Section 2.4), we focus on large-scale time series aggregates. Our attack is cast as a time series classification problem. We leverage the temporal aspect of our data by using specialized time series classifiers. As the data changes over time (due to meteorology and human interaction), we evaluate the effectiveness of domain adaptation methods to improve the attack success rate. We evaluate the attack's success according to the aggregate size, the series length, and the target. Our findings demonstrate the vulnerability of large aggregates (up to 20,000 series) against favorable adversaries performing the attack during the same period as the training. The attack is less successful against weaker attackers performing the attack during a different period than the training, even if we still manage to attack aggregates of size 1000. We provide the oddness score as a metric that can estimate individual series' potential vulnerability to membership inference attacks. Eventually, our results will help publishers disclose smaller aggregates with better privacy guarantees.

We organize this chapter as follows: Section 5.1 contains the necessary background knowledge about time series classification. Section 5.2 introduces the context of the attack, the publishing environment, and the considered threat model. Section 5.3 presents the algorithms of the attack. Section 5.4 presents the experimental protocol. We perform an in-depth study of the attack and the results in Section 5.5. Section 5.6 finally concludes this chapter.

5.1 Background knowledge

Our attack is cast as a time series classification problem. Time series are specific data types with strong correlations between data points. [Bag+16; MSB23] [1] surveys and bench-

1. `http://timeseriesclassification.com`

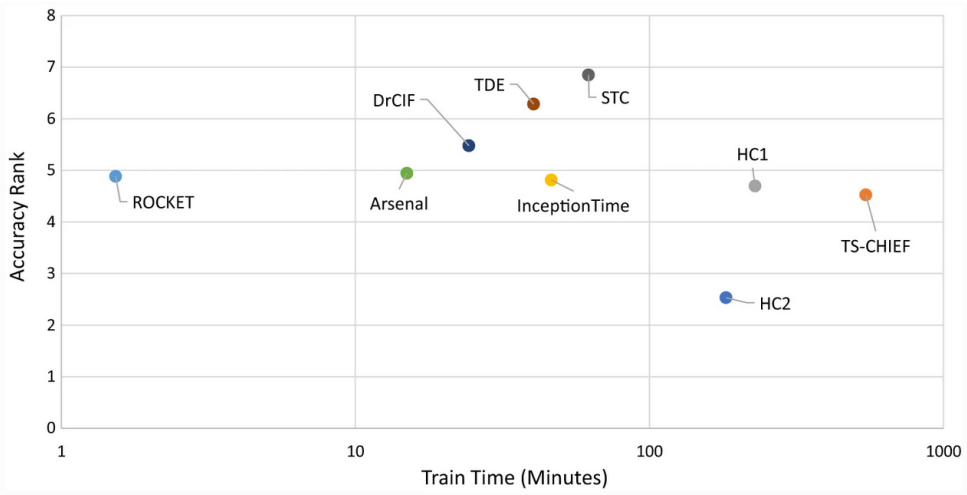


Figure 5.1 – Comparaison of training time versus model accuracy rank for various time series classification models (from [Mid+21]).

mark the different methods to classify time series. Figure 5.1 (from [Mid+21]) compares the training time versus the accuracy rank of the current state-of-the-art time series classification algorithms. The rank is computed using the Wilcoxon signed rank test [Wil45]. As we plan to perform many experiments on large-scale datasets, we also pay particular attention to the model's computational cost. While HIVE-COTE V2 [Mid+21] (HC2 on the figure) offers the best accuracy, Rocket [DPW20] offers the best tradeoff between accuracy and computation time. The creators of Rocket propose MiniRocket [DSW21], an improvement of Rocket. According to the authors, MiniRocket is 75 times faster than Rocket and offers the same accuracy. Therefore, we use MiniRocket to perform our experiments.

MiniRocket computes a set of random convolutional kernels [2] for every training series (in our case, an aggregate). The kernels are used as features in a linear model to classify the data. By default, Rocket and MiniRocket use a Ridge classifier [HK70] but it is possible to use a Logistic Regression classifier [Cra02]. The only hyperparameter is the number of kernels. All aspects (length, weight, bias, dilatation, and padding) of the kernels are random. The linear classifier takes as features the PPV (Proportion of Positive Values) of each kernel. Equation 5.1 defines the PPV for the series X, the kernel K, and the bias b.

2. A convolutional kernel is a small matrix transforming each point of the time series depending on its neighbors.

$$PPV(X * K) = \frac{1}{2} \sum [X * K > b] \tag{5.1}$$

The classifier produces a **confusion matrix** describing the proportion of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) examples classified by the model. The confusion matrix is used to compute the following metrics evaluating the model performance:

Accuracy Formulae 5.2 defines the accuracy as the ratio of correct classification (both positive and negative) during the testing phase. In a binary classification setting, an accuracy close to 1 indicates that the model mainly makes correct classifications. An accuracy score close to 0.5 means that the model cannot make good predictions and is answering randomly.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.2}$$

Precision Formulae 5.3 defines the precision as the proportion of correct positive classifications. A precision close to 1 indicates that the model can confidently identify positive examples but may produce many false negatives. A precision close to 0 indicates that the model makes many false positive predictions.

$$precision = \frac{TP}{TP + FP} \tag{5.3}$$

Recall Formulae 5.4 defines recall as the proportion of positive labels correctly identified by the model. A recall close to 1 indicates that the model can identify many positive examples, but it may also produce many false positives. A recall close to 0 indicates that the model is missing many positive examples and may produce many false negatives.

$$recall = \frac{TP}{TP + FN} \tag{5.4}$$

F-score Formulae 5.5 defines the F-score as harmonic mean of the precision and the recall. It evaluate the model capacity to perform correct positive classifications. An F-score close to 1 indicates that the model has both a high precision and a high recall, while an F-score close to 0 indicates that the model has poor precision or recall scores.

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5.5}$$

Domain Adaptation: Domain Adaptation is a set of machine learning techniques that can be used when the training data distribution (called the source) is different from the testing data distribution (called the target). domain adaptation algorithms either transform the data from a source to a target domain or directly transform the model [Red+20; Zha21]. There exist supervised and unsupervised domain adaptation algorithms. We focus on unsupervised algorithms that transform the data since we use these algorithms in a context where we cannot access target labels and cannot modify the model. The Adapt Python library [3] references and implements multiple state-of-the-art domain adaptation methods. Three algorithms match our constraints (i.e., unsupervised data transformation): Transfer Component Analysis (TCA) [Pan+09], Subspace Alignment (SA) [Fer+13] and CORAL [SFS17]. We could not run the TCA algorithm on our server due to a lack of RAM (the server has 1 TB of RAM), so we focus on SA and CORAL.

Subspace Alignment (SA) [Fer+13] is defined in Algorithm 2. It adapts the source domain to the target one by aligning the two PCA vectors [Hot36]. By default, the number of components is set to the dimension of the dataset. With time series, it corresponds to the length of the series. Note that, in the algorithm X_S' is the transposed matrix of X_S, the source PCA vector.

Algorithm 2: Subspace Alignment (SA)

Input: The source (S^{source}) and target (S^{target}) datasets and the number of PCA compoments (d).
Output: The transformed source dataset (S_T^{source})

$X_S = PCA(S^{source}, d)$
$X_T = PCA(S^{target}, d)$
$X_a = X_S \cdot X_S' \cdot X_T$
$S_T^{source} = S^{source} \cdot X_a$
return S_T^{source}

CORAL [SFS17] aligns the covariance matrix between the source and the target domain using Algorithm 3.

3. `https://adapt-python.github.io/adapt/map.html`

Algorithm 3: CORAL

Input: The source (S^{source}) and target (S^{target}) datasets.
Output: The transformed source dataset (S_T^{source})

$X_S = cov(S^{source}) + I_{size(S,2)}$
$X_T = cov(S^{target}) + I_{size(T,2)}$
$S = S * X_S^{-1/2}$
$S_T^{source} = S * X_T^{1/2}$
return S_T^{source}

DTW: Algorithm 4 defines the **D**ynamic **T**ime **W**arping (DTW) between two time series S_i and S_j of length $|T_i|$ and $|T_j|$. DTW is an algorithm measuring the similarity between two time series. A DTW value close to 0 means that the two series are similar.

Algorithm 4: DTW

Input: S_i and S_j, two time series of length n and m.
Output: The DTW distance.

$$D = \begin{bmatrix} \infty_{0,0} & \cdots \\ \vdots & \infty_{n,m} \end{bmatrix}$$

for $a = 1;\ a < n;\ a++$ **do**
  for $b = 1;\ b < m;\ b++$ **do**
    cost = $|S_{i,a} - S_{j,b}|$
    insert = $D[a-1;\ b]$
    delete = $D[a;\ b-1]$
    match = $D[a-1;\ b-1]$
    $D[a;b]$ = cost + min([insert, delete, match])
  end
end
return $D[n;m]$

5.2 The publishing environment

The publishing environment comprises a publisher and an attacker. The publisher collects a set of time series $\mathcal{S}^{pub}$ from a set of individuals $\mathcal{I}$. Each individual produces a single fixed-length time series associated with additional, possibly private, attributes contextualizing the time series. Table 5.1 summarizes the notations used in this chapter.

Notation	Description
$\mathcal{I}$	Set of individuals
$\mathcal{T}$	Set of timestamps
$\mathcal{S}^{pub} \subset \mathcal{S}$	Publisher's series
$\mathcal{S}^{\mathcal{A}} \subset \mathcal{S}^{pub}$	Aggregate's series
$c \in \mathcal{S}^{pub}$	Targeted series
$\mathcal{S}^{att} \subset \mathcal{S}$	Attacker's series
$\mathcal{A}$	Published aggregates
$\mathcal{O}$	Outlier score
$\mathcal{M}$	Classification model
accuracy	Model accuracy score
$\lvert \cdot \rvert$	Cardinality

Table 5.1 – Notations

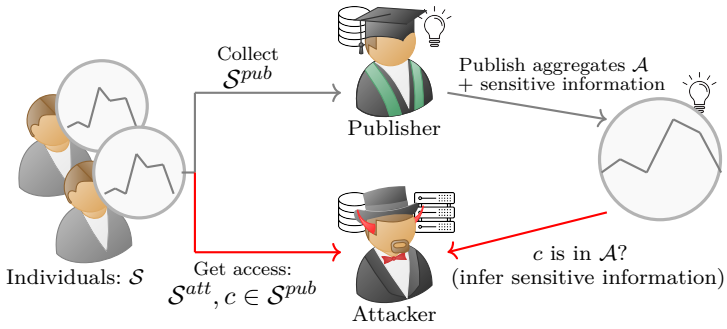


Figure 5.2 – The *STATS* Attack

The publisher: The publisher selects a subset $\mathcal{S}^{\mathcal{A}} \subset \mathcal{S}^{pub}$, computes and publishes a vector of aggregates $\mathcal{A}$ from $\mathcal{S}^{\mathcal{A}}$. This chapter focuses on aggregates that are **means** (we use the two terms indistinctly below). Note that our attack can also tackle all sum-based aggregation methods (e.g., sums and means) without any changes to the attack. It is worth noting that the vector of aggregates $\mathcal{A}$ usually comes along with the timestamps and the number of time series aggregated. Formulea 5.6 computed the aggregate $\mathcal{A}_t$ per timestamp with $S^{\mathcal{A}}_{i,j}$ the consumption of the i^{th} individual at the timestamp t.

$$\mathcal{A}_t = \frac{\sum_{\forall i} \mathcal{S}^{\mathcal{A}}_{i,t}}{|\mathcal{S}^{\mathcal{A}}|} \tag{5.6}$$

The attacker: The threat model (illustrated in Figure 5.2) considers an attacker aiming to infer the presence of a known target inside a published aggregate. During the attack, the attacker has access to the series of a targeted individual c and a set of series S^{att} similar

to the ones at the origin of the aggregates. Simimar series are not formally defined. They are similar if they share similar statistical properties (i.e., shape and frequency). Even if the attacker series S^{att} could intersect with the publisher series S^{pub}, we consider the worst-case scenario where S^{att} is distinct from $\mathcal{S}^{pub}$. The attacker could readily obtain similar series using legal means such as data-sharing programs or open data. Getting the precise series of the target might seem complex, yet several actors have such access, using legal means (e.g., former clients) or illegal ones (e.g., data leaks). The attacker can infer information by finding whether the target is part of the aggregate. For example, the simple knowledge of the target presence can help discriminate between a client that moved (absent from the newest publications) and a client that is gone with a competitor (still present in the newest publications). Secondly, as the aggregate is published with private attributes, knowing the presence of the target within the aggregate allows labeling the target with this information.

We distinguish two variants of the attack: when the attacker has access to c and S^{att} over the same period as the publication and when he has access to information over a different period. The latest is the worst-case scenario for the attacker and the most realistic one. We do not rule out an attacker getting access to his targeted series during the same period as the attacked one. In this case, the attacker's motivation declines (without canceling it) as much information is disclosed, but it eases his task. Considering the two scenarios bound the attack by its extreme. We explore both the easiest and the worst setup for the attacker.

5.3 The STATS Attack

The *STATS* attack is cast as a time series classification problem. The attacker uses a set of known series $\mathcal{S}^{att}$, similar to the ones used by the publisher, to generate fake aggregates. The fake aggregates are balanced, containing the same number of aggregates with the target as without it. The generation of fake aggregates to train the attack model is called **shadow training**. The classifier is trained using the fake aggregate to detect the presence of the target. The classifier is used against the targeted aggregate to infer the target membership with a probability depending on the model's accuracy.

Algorithm 5 presents the attack algorithm against an aggregate of size $|\mathcal{S}^{\mathcal{A}}|$. It uses the targeted series c and three, ideally distinct (population and period wise), sets of series: $\mathcal{S}^{train}$, and respectively $\mathcal{S}^{valid}$ and $\mathcal{S}^{test}$, subset of S^{att}, to train, validate and test the

Algorithm 5: *STATS* attack

Input: The targeted series (c), The set of series known to the attacker for training ($\mathcal{S}^{train}$), for validation ($\mathcal{S}^{valid}$) and testing ($\mathcal{S}^{test}$), The number of train (k^{train}), validation (k^{valid}) test (k^{test}) aggregates, Aggregate size ($|\mathcal{S}^{\mathcal{A}}|$)

Output: The classification model (f) and its confusion matrix ($\mathcal{M}^{valid}$ and $\mathcal{M}^{test}$).

`// Build train / test aggregates`
$\mathcal{X}^{train}$, $\mathcal{Y}^{train}$:= *build_aggregates*($\mathcal{S}^{train}$, c, k^{train}, $|\mathcal{S}^{\mathcal{A}}|$)
$\mathcal{X}^{valid}$, $\mathcal{Y}^{valid}$:= *build_aggregates*($\mathcal{S}^{valid}$, c, k^{valid}, $|\mathcal{S}^{\mathcal{A}}|$)
$\mathcal{X}^{test}$, $\mathcal{Y}^{test}$:= *build_aggregates*($\mathcal{S}^{test}$, c, k^{test}, $|\mathcal{S}^{\mathcal{A}}|$)
`// (Optional) Domain adaptation`
$\mathcal{X}^{train}$:= *adapt*($\mathcal{X}^{train}, \mathcal{X}^{test}$)
`// Train and test the model`
f = *train_model*($\mathcal{X}^{train}$, $\mathcal{Y}^{train}$)
$\mathcal{M}^{valid}$:= *test_model*(f, $\mathcal{X}^{valid}$, $\mathcal{Y}^{valid}$)
$\mathcal{M}^{test}$:= *test_model*(f, $\mathcal{X}^{test}$, $\mathcal{Y}^{test}$)
return f, $\mathcal{M}^{valid}$, $\mathcal{M}^{test}$

Functions used in the algorithm :

build_aggregates($\mathcal{S}$, c, k, $|\mathcal{S}^{\mathcal{A}}|$)**:** Build a set of k aggregates of size $|\mathcal{S}^{\mathcal{A}}|$ from a series dataset $\mathcal{S}$ containing the target c and another k aggregates not containing the target c (Algorithm 6).

adapt($\mathcal{X}^{training}, \mathcal{X}^{testing}$)**:** Domain adaptation algorithm transforming the source aggregates (training) to match the destination aggregates distribution (testing).

train_model($\mathcal{X}$, $\mathcal{Y}$)**:** Train the classifier using a set of aggregates ($\mathcal{X}$) and a target presence vector ($\mathcal{Y}$).

test_model(f, $\mathcal{X}$, $\mathcal{Y}$)**:** Test the model against a set of aggregates ($\mathcal{X}$) and the target presence ground truth ($\mathcal{Y}$).

model. Firstly, the attacker uses Algorithm 6 to generate three sets of aggregates ($\mathcal{X}^{train}$) of size $|\mathcal{S}^{\mathcal{A}}|$ for training, validation, and testing using their respective population. The symbol $\frown$ denotes the concatenation of two vectors. All sets of aggregates are balanced and contain k aggregates including the target c and k aggregates without c. A boolean vector ($\mathcal{Y}$) label which aggregates contains the target. Once the aggregates are generated, the training aggregates ($\mathcal{X}^{train}$) and label vector ($\mathcal{Y}^{train}$) are used to train the model to detect the target presence. Subsequently, the model is tested against the validation and testing aggregate ($\mathcal{X}^{valid}$, $\mathcal{X}^{test}$) and label vector ($\mathcal{Y}^{valid}$, $\mathcal{Y}^{test}$) and returned alongside the resulting confusion matrix $\mathcal{M}$, containing TP, TN, FP, FN respectively the number of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) classifications. The confusion matrix is used to compute the following metrics: accuracy, precision, recall, and the F-score. If the model performs correctly, the model is used against the attacked aggregate, inferring the target presence with a correctness probability depending on the metric observed.

Algorithm 6: build_aggregates

Input: The series dataset ($\mathcal{S}$), The targeted series (c), The number of aggregates (k), The aggregate size ($|\mathcal{S}^{\mathcal{A}}|$).

Output: The set of aggregates ($\mathcal{X}$) and the target presence vector ($\mathcal{Y}$).

$\mathcal{X} = []$
$\mathcal{Y} = []$
for $i = 0$ **to** k **do**
  $\mathcal{S}^{c} := \mathcal{S} \setminus \{c\}$
  $\mathcal{S}^{base} := sample(\mathcal{S}^{c},\ |\mathcal{S}^{\mathcal{A}}| - 1)$
  $\mathcal{S}^{with} := \mathcal{S}^{base} \cup \{c\}$
  $\mathcal{S}^{without} := \mathcal{S}^{base} \cup sample(\mathcal{S}^{c} \setminus \mathcal{S}^{base}, 1)$
  $\mathcal{A}^{with} := aggregate(\mathcal{S}^{with})$
  $\mathcal{A}^{without} := aggregate(\mathcal{S}^{without})$
  $\mathcal{X} := \mathcal{X} \frown \left[\mathcal{A}^{with};\ \mathcal{A}^{without}\right]$
  $\mathcal{Y} := \mathcal{Y} \frown [1;\ 0]$
end
return $\mathcal{X},\ \mathcal{Y}$

Functions used in the algorithm :

***sample*($\mathcal{S}$, n):** Randomly sample n series from the series dataset $\mathcal{S}$.

***aggregate*($\mathcal{S}$):** Aggregate (sum or mean) the set of series $\mathcal{S}$ per timestamp.

Alternatively, the attack can use a domain adaptation algorithm to boost its results. Due to technical limitations [4] related to the classifier and the domain adaptation algorithms, the algorithms adapt the aggregates. They transform the training aggregates to resemble the testing ones.

5.4 Experiments

Our experiments aim at understanding which parameters influence the success of our attack. We consider a list of parameters impacting the results: the aggregate size $|\mathcal{S}^{\mathcal{A}}|$, the series length $|\mathcal{T}|$, the target c, and the train/test population and period. For each set of parameters, we apply the algorithm defined in Section 5.3 building $k^{train} = 15,000$ training aggregates pairs and $k^{valid} = k^{test} = 5,000$ validation and testing aggregates. The testing aggregates challenge the model to obtain the model performance metrics (accuracy, precision, recall, and F-score). We consider a target **vulnerable** for a series length and aggregate size if its accuracy is above 0.6 meaning the attacker can gain some advantage over random guessing. The validation phase is used before the experiments to fix the model hyperparameters (i.e., rocket kernels and normalization).

We propose an open-source Python implementation of the *STATS* attack allowing the reproduction of our results [5]. We deploy our experiments on a computer with 32 cores (2.5 Ghz) and 1 TB of RAM. We use the minirocket [DSW21] classifier with 1,000 kernels as it offers the best tradeoff between accuracy and classification time.

5.4.1 Datasets

We perform the attack on two real-life electric consumption datasets: the *CER-ISSDA* presented in Section 3.2 and the *ENEDIS* dataset presented in Section 3.4. Due to the small size of the *ISSDA* dataset, we do not split the population into separate populations, and the targets are selected directly from this population.

The *ENEDIS* dataset contains around 1.6 million residential series (RES1 and RES2) over two months (June 2021 and June 2022) with a half-hourly frequency and precision to the Watt. Despite being collected after the *ENEDIS* dataset in Section 3.4, this dataset has the same statistical properties. Each profile has a different consumption pattern. We keep the balance between each category in the following series selection. We split the *ENEDIS*

4. The implementation of Rocket we use is incompatible with our domain adaptation library.
5. `https://gitlab.com/phd_antonin/mia-ts`

dataset into 4 datasets: First, we select the targets as defined in Section 5.4.2. Then, we extract three (*ENEDIS-train*, *ENEDIS-valid*, and *ENEDIS-test*, used respectively as training, validation, and testing populations) distinct representative panels. The *ENEDIS-train* dataset contains 100,000 series. The *ENEDIS-valid* and *ENEDIS-test* contain 50,000 series each. A non-negligible amount of series (up to 4 % of the series per timestamp) contains at least one missing value over the period. We select the most complete series and interpolate the missing values (using linear interpolation) when the missing period per day is less than 1H long. While most selected series are complete, around 2 % of them require interpolation with, at worst, 2.5 % (36 timestamps over 1440) missing timestamps in them.

5.4.2 Targets selection

We select the targets by attributing an **oddness score** $\mathcal{O}_s$ to each series. This score tells how far the series is from the **mean-series**. The mean-series is the mean aggregate of all the series within the whole dataset (*ISSDA* or *ENEDIS*). Formulae 5.7 computes the score $\mathcal{O}_s$ for each series $s \in \mathcal{S}$. Formulae 5.8 computes the mean-series μ_t for timestamp t.

$$\forall i \in \mathcal{I}, \mathcal{O}_i = \frac{\sqrt{\sum_{t \in T}(\mu_t - \mathcal{S}_{i,t})^2}}{|T|} \tag{5.7}$$

$$\mu_t = \frac{\sum_{\forall i \in \mathcal{S}} \mathcal{S}_{i,t}}{|\mathcal{S}_t|} \tag{5.8}$$

A high score corresponds to a large dispersion regarding the mean-series. Targets are selected according to the score. The score domain is split into multiple groups (defined below), and the targets are sampled within each group. We define the groups as giving more importance to the series with a more significant score that would otherwise not be selected by uniformly sampling the whole population. We call these series **outliers** as they are few and far from the average series. We expect outliers to be more vulnerable than the series close to the mean-series. As they are few and different from the other series, we expect them to produce a stronger impact on the aggregates. However, the groups are strongly imbalanced, with some representing around 90 % of the total population and others representing only 1 %.

ISSDA: We split *ISSDA* dataset into 4 groups of equal size relative to the oddness scores and the standard deviation of the score $\sigma(\mathcal{O})$. From them, we sampled 10 series per group when possible. Groups are defined in the following manner:

- ***G0*:** $[0; 1 \times 5 \times \sigma(\mathcal{O})]$. It contains 4562 series (98 %).
- ***G1*:** $]1 \times 5 \times \sigma(\mathcal{O}); 2 \times 5 \times \sigma(\mathcal{O})]$. It contains 49 series (1.06 %).
- ***G2*:** $]2 \times 5 \times \sigma(\mathcal{O}); 3 \times 5 \times \sigma(\mathcal{O})]$. It contains 7 series (0.1 %).
- ***G3*:** $]3 \times 5 \times \sigma(\mathcal{O}); \max(\mathcal{O})]$. It contains 3 series (0.06 %).

Figure 5.3 shows the distribution of the series score (the histogram in blue), the groups (the vertial lines in red), and the score of the target (the dots in orange). We force the selection of significant outliers that would not have been selected by uniform sampling. However, the most extreme groups (*G2* and *G3*) contain a tiny number of series. Therefore all series in these groups are selected as targets.

ENEDIS: The Enedis dataset is split into three groups around the average score. We reduced the number of groups compared *ISSDA* due to the few numbers of individuals present in G2 and G3. This selection method allows watching three series categories: the series close to the mean series, the main population, and the outliers. We selected 100 series per group and profiles as targets. Groups are defined in the following manner:

- ***G0*:** contains series with a lower score than *G1*. $G0 = [0; \overline{\mathcal{O}} - \sigma(\mathcal{O})]$. It contains 23,039 series (4.74 %) for the *RES1* and 10,080 (1.14 %) for the *RES2*.
- ***G1*:** contain series around. $G1 =]\overline{\mathcal{O}} - \sigma(\mathcal{O}); \overline{\mathcal{O}} + \sigma(\mathcal{O})]$. It contains 421,520 series (86.75 %) for the *RES1* and 781,829 (89.09 %) for the *RES2*.

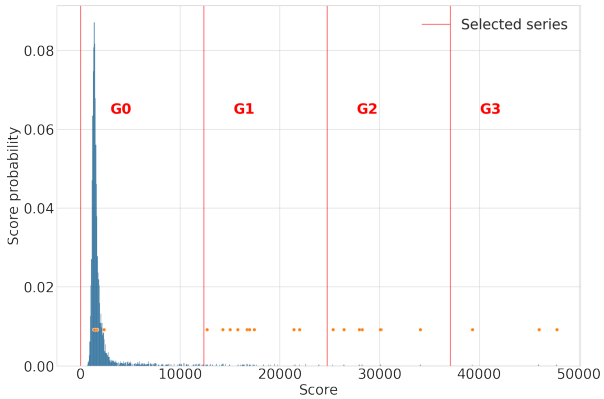
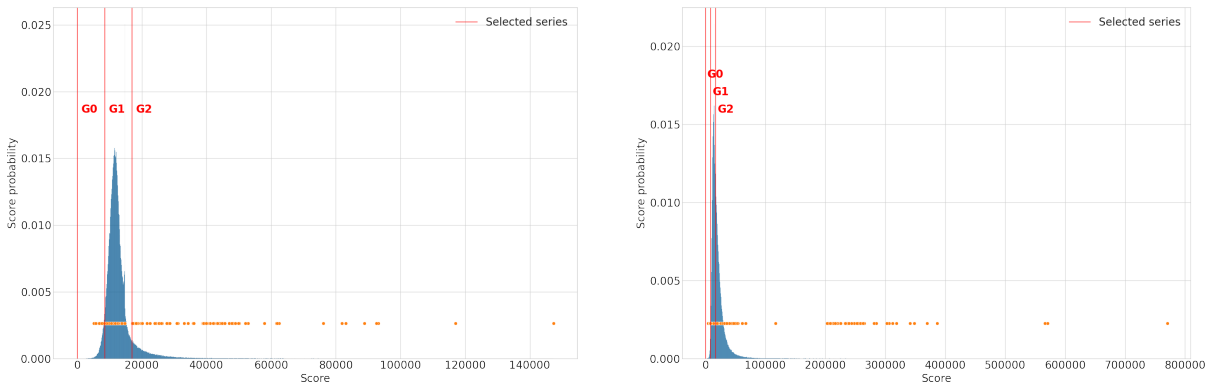


Figure 5.3 – ISSDA score



(a) RES1 (b) RES2

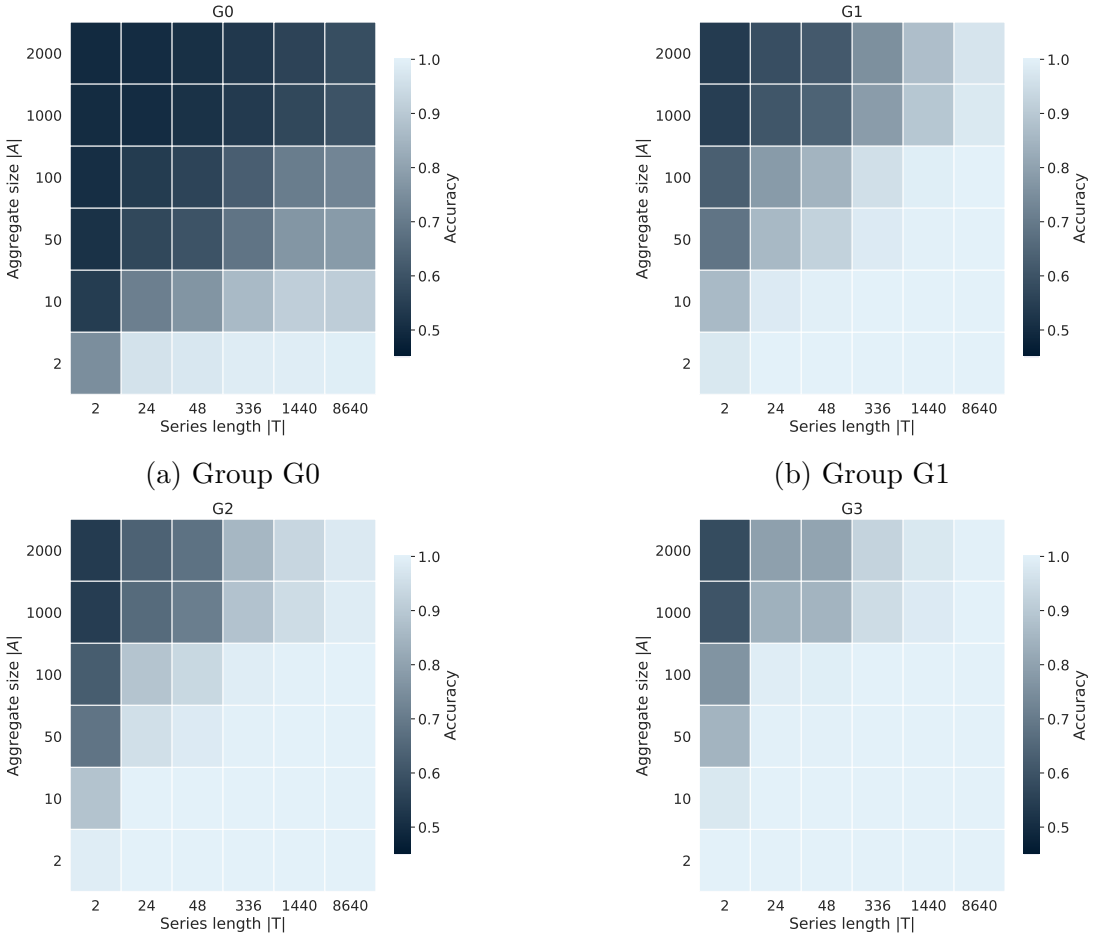
Figure 5.4 – ENEDIS score

— ***G2*:** contains series with a higher score than *G1*. $G2 =]\overline{\mathcal{O}} + \sigma(\mathcal{O}); \max(\mathcal{O})]$. It contains 41,303 series (8.5 %) for the *RES1* and 85,629 (9.75 %) for the *RES2*.

Individual series scores are computed against the population (1.6 M series), and targets are selected relative to the profiles series distribution. Figure 5.4 shows the series score distribution (histogram in blue) for each profile and the score of each targeted series (orange dots). Vertical red lines delimitate the groups. *RES1* (Figure 5.4a) series are generally more dispersed than the *RES2* series (Figure 5.4b), with most series having a score around 10,000 for the RES1 compared to 2500 for the RES2. Nevertheless, *RES2* series have more extreme outliers with a maximum score of 780,000, 5 times higher than *RES1* series (150,000).

5.5 Experimental results

In this section, we analyze the results of the experiments. First, we observe the accuracy on the *ISSDA* dataset function of the aggregate size, group, and series length. Then, we observe the accuracy on the *ENEDIS* dataset. We start by performing the attack in the same period as the training period (June 2021). We refer to this scenario as "same period". Then, we observe the accuracy by training the classifier one year prior to the attack period. The training uses June 2021 data, and the testing uses June 2022 data. We refer to this scenario as "different period". We test the impact of domain adaptation algorithms on the attack's performances.



(a) Group G0

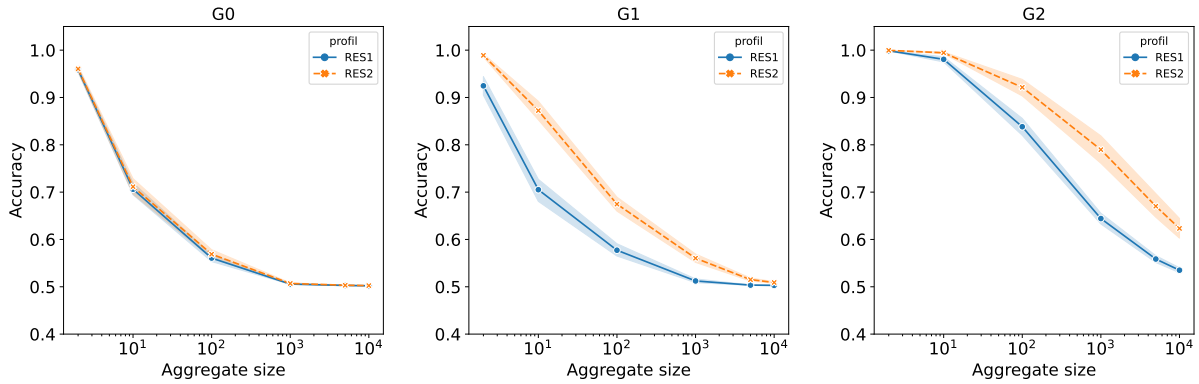
(b) Group G1

(c) Group G2

(d) Group G3

Figure 5.5 – STATS attack accuracy on the *ISSDA* dataset depending on the aggregate size, series length, and score (group). $|\mathcal{S}^{\mathcal{A}}| = \{2, 10, 50, 100, 1000, 2000\}$, $|\mathcal{T}| = \{2, 24, 48, 336, 1440, 8640\}$.

Attacking the *ISSDA* dataset: Figure 5.5 shows the models accuracy on the ISSDA dataset according to the aggregate size $|\mathcal{S}^{\mathcal{A}}|$ and the series length $|\mathcal{T}|$ for each target group. We see a clear impact of both the aggregate size $|\mathcal{S}^{\mathcal{A}}|$ and the series length $|\mathcal{T}|$ on the accuracy. Small aggregates and long series are the most vulnerable (accuracy close to 1). Large and short aggregates are almost impossible to attack (accuracy below 0.6). The oddness score influences the attack accuracy. Distinctive targets (*G3*, *G2* and *G1*) are more vulnerable considering larger aggregates and shorter series. Targets from the most common series (i.e., *G0*) are harder to distinguish, even considering small and long aggregates.

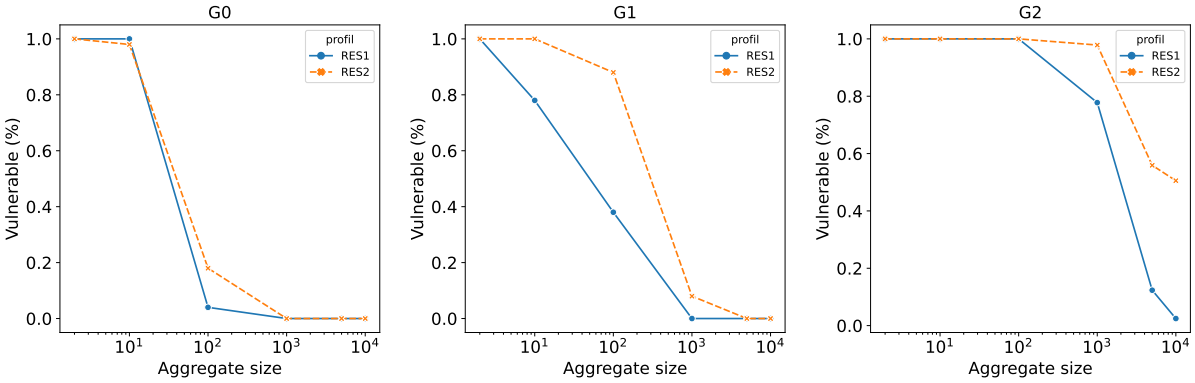


(a) Group G0 (b) Group G1 (c) Group G2

Figure 5.6 – STATS attack accuracy on the *ENEDIS* dataset depending on the aggregate size, profile, and group over the same period. $|\mathcal{S}^{\mathcal{A}}| = \{2, 100, 1000, 5{,}000, 10{,}000\}$, $|\mathcal{T}| = 1440$, training = 2021, testing = 2021.

Attacking *ENEDIS* aggregates: Figure 5.6 shows the average accuracy (as well as the 95 % confidence interval) on the *ENEDIS* dataset according to the aggregate size, group, and profile. Both training and testing are performed in the same period (June 2021). As in the previous figures, large aggregates are more challenging to attack than small ones. Secondly, we see that *RES2* series are more vulnerable than the *RES1* ones. Lastly, the target group influences the resulting accuracy. For the *G0* series, the model's accuracy is below 0.6 for the aggregate size of $|\mathcal{S}^{\mathcal{A}}| > 100$ ($|\mathcal{S}^{\mathcal{A}}| > 1,000$ for *G1*) while we still have a decent accuracy (0.65) for the RES2 G2 series at $|\mathcal{S}^{\mathcal{A}}| = 10,000$.

Figure 5.7 shows the proportion of targeted individuals vulnerable (i.e., with an accuracy > 0.6) on the *ENEDIS* dataset according to the aggregate size, group, and profile.



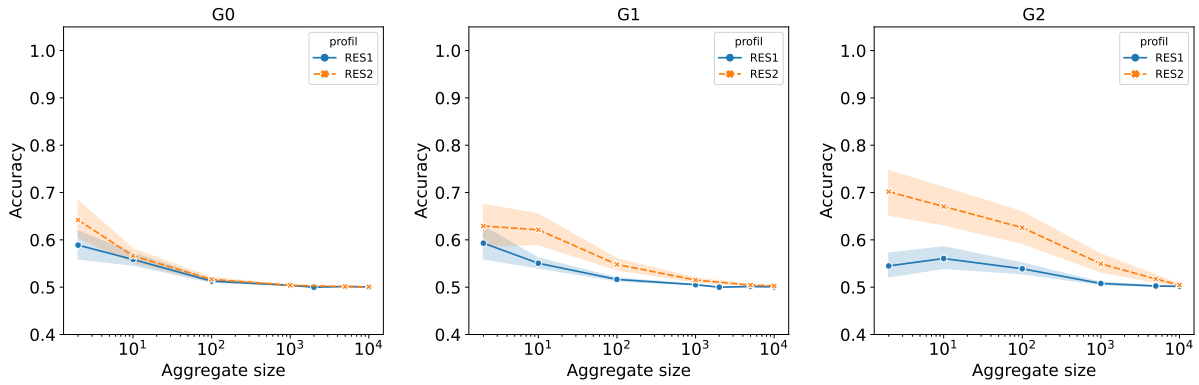
(a) Group G0 (b) Group G1 (c) Group G2

Figure 5.7 – Proportion of vulnerable individuals on the *ENEDIS* dataset depending on the aggregate size, profile, and group over the same period. $|\mathcal{S}^{\mathcal{A}}| = \{2, 100, 1000, 5000, 10000\}$, $|\mathcal{T}| = 1440$, training = 2021, testing = 2021.

The proportion of vulnerable individuals diminishes as the aggregate size increases. Individuals are vulnerable when the aggregate size is small (below $|A| \leq 10$ for G1 and $|A| \leq 100$ for G2). For group G0, an aggregate size of $|A| = 1,000$ is enough to protect all individuals. However, this threshold is reached at $|A| = 10,000$ for the group G2 and the profile RES1. We note that approx. 50 % RES2 in the group G2 are still vulnerable for an aggregate size of $|A| = 10,000$. At most, we attack aggregates of 20,000 series, and 22 % of the RES2 G2 series remain vulnerable with a maximal accuracy of 0.77.

Attacking different period: Figure 5.8 shows the mean attack accuracy (as well as the 95 % confidence interval) per profile and group according to the aggregate size with different training (June 2021) and test (June 2022) period. As the aggregate size increases, the attack accuracy for all targeted series diminishes towards 0.5. *RES1* series generally have a very low accuracy (below 0.7 for $|\mathcal{S}^{\mathcal{A}}| = 2$). However, we observed individual series in the *G2* group reaching an accuracy of 0.8 at an aggregate size of $|\mathcal{S}^{\mathcal{A}}| = 100$. *RES2* series have an higher average accuracy than the *RES1* series reaching 0.65 at $|\mathcal{S}^{\mathcal{A}}| = 100$. We observe a maximum accuracy of 0.75 for $|A| = 5,000$. In France, cold weather is one of the main factors influencing electricity consumption. By choosing two summer periods, we limit the impact of the weather on the series. However, the series changes from one year to the next. Therefore, series are harder to classify, resulting in a lower attack accuracy.

Figure 5.9 shows the proportion of vulnerable targets (i.e., with an accuracy > 0.6) on the *ENEDIS* dataset according to the aggregate size, group, and profile. The proportion



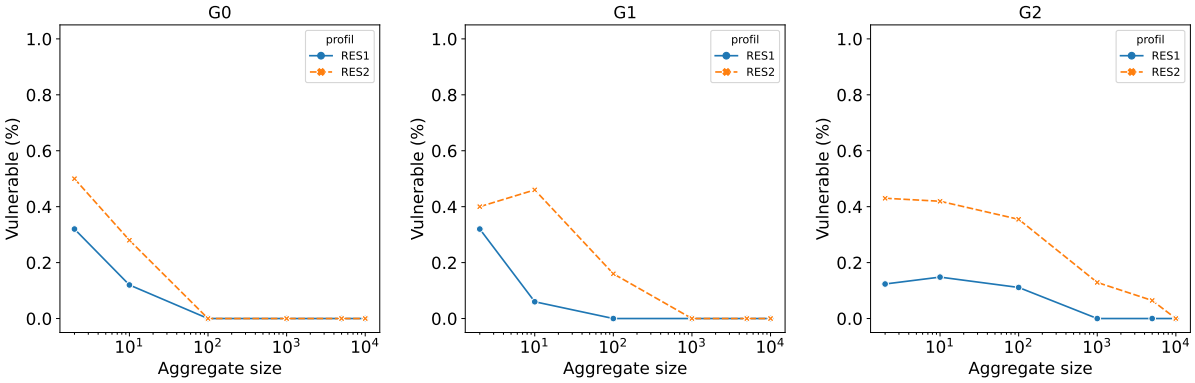
(a) Group G0 (b) Group G1 (c) Group G2

Figure 5.8 – STATS attack accuracy on the *ENEDIS* dataset depending on the aggregate size, profile, and group over a different period. $|\mathcal{S}^{\mathcal{A}}| = \{2, 100, 1000, 5000, 10000\}$, $|\mathcal{T}| = 1440$, training = 2021, testing = 2022.

of vulnerable targets follows a similar shape as the attack in the same period, except the proportion is half. At most, approx. 50 % individual are vulnerable for $|\mathcal{S}^{\mathcal{A}}| = 2$. For group G0, no individual is vulnerable for $|\mathcal{S}^{\mathcal{A}}| \geq 100$. For group G1 and the RES1 profile of group G2, no individual is vulnerable for $|\mathcal{S}^{\mathcal{A}}| \geq 100$. For the RES2 profile of group G2, no individuals are vulnerable for $|\mathcal{S}^{\mathcal{A}}| = 10,000$, but 10 % remain vulnerable for $|\mathcal{S}^{\mathcal{A}}| = 5,000$.

Impact of the domain adaptation methods: Figure 5.10 and Figure 5.11 show the mean accuracy using the CORAL (and respectively the Subspace Alignment) domain adaptation method. CORAL improves the attack accuracy by up to 40 % for aggregates size below $|\mathcal{S}^{\mathcal{A}}| \leq 100$. We do not observe any significant improvements for aggregates size above $|\mathcal{S}^{\mathcal{A}}| > 100$. The usage of the SA method does not significantly improve the accuracy (at most +10 % for $|\mathcal{S}^{\mathcal{A}}| = 2$).

Figure 5.12 and Figure 5.13 show the proportion of vulnerable targets using the CORAL (and respectively SA) domain adaptation method. Both CORAL and SA significantly improve the proportion of vulnerable targets for small aggregate sizes ($|\mathcal{S}^{\mathcal{A}}| < 100$). We observe no improvement for larger aggregate sizes. CORAL is the most efficient domain adaptation method improving the proportion of vulnerable targets by up to 80 % for RES1 G2 on $|\mathcal{S}^{\mathcal{A}}| = 2$. At most, SA improves the proportion of vulnerable targets by at most 40 % for RES1 G2 on $|\mathcal{S}^{\mathcal{A}}| = 2$. However, for aggregates size above $|\mathcal{S}^{\mathcal{A}}| > 2$, SA



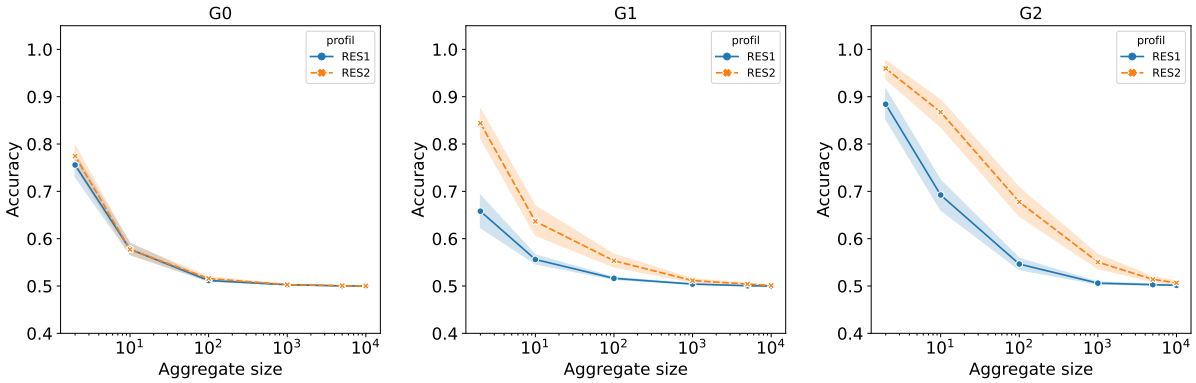
(a) Group G0 (b) Group G1 (c) Group G2

Figure 5.9 – Percentage of vulnerable individuals on the *ENEDIS* dataset depending on the aggregate size, profile, and group over a different period. $|\mathcal{S}^{\mathcal{A}}| = \{2, 100, 1000, 5000, 10000\}$, $|\mathcal{T}| = 1440$, training = 2021, testing = 2022.

does not improve the proportion of vulnerable targets. Sometimes SA even reduces the proportion of vulnerable targets.

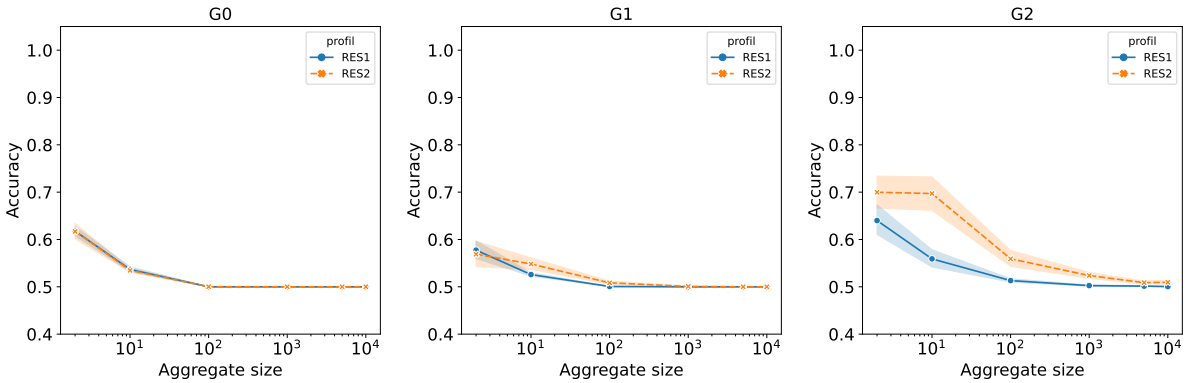
Explaining the accuracy drop attacking different periods: Figure 5.14 shows the accuracy change between the attack on the same period and the attack on a different period depending on the DTW distance, the score's drift, and the series Euclidian distance. The DTW is the Dynamic Time Warping distance between the two series. The score drift is defined as the difference between the series score in the training period and in the testing period. The distance is the Euclidian distance between the two series. Performing the attack during a different period always degrades the accuracy compared to attacking the same period. The accuracy loss is lower when all parameters are close to 0. The more the target changes between the training and testing periods, the more challenging the attack. Note that the accuracy change is sometimes low when the distance is relatively high because the accuracy was already close to 0.5.

Other success metrics: Besides the accuracy, we look at the attack results using the precision, recall, and F-score metrics. Figure 5.15 shows the attack performance using these metrics. When attacking the same period, all metrics follow the accuracy. The precision follows the accuracy on different periods and diminishes with the aggregate size. The recall remains stable or increases with the aggregate size. The F-score remains stable (around 0.5) or increases with the aggregate size. Using domain adaptation methods



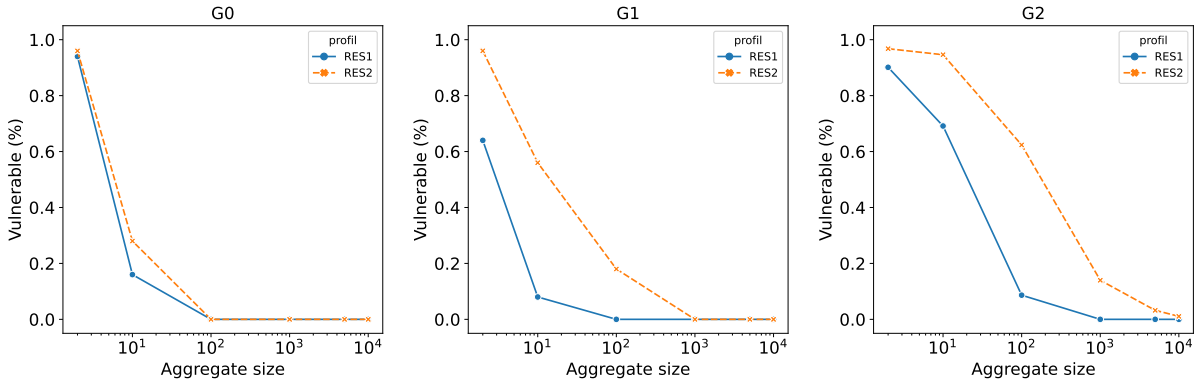
(a) Group G0 (b) Group G1 (c) Group G2

Figure 5.10 – STATS attack accuracy on the *ENEDIS* dataset per target and profile over a different period using the CORAL domain adaptation. $|\mathcal{S}^{\mathcal{A}}| = \{2, 100, 1000, 5000, 10000\}$, $|\mathcal{T}| = 1440$, training = 2021, testing = 2022.



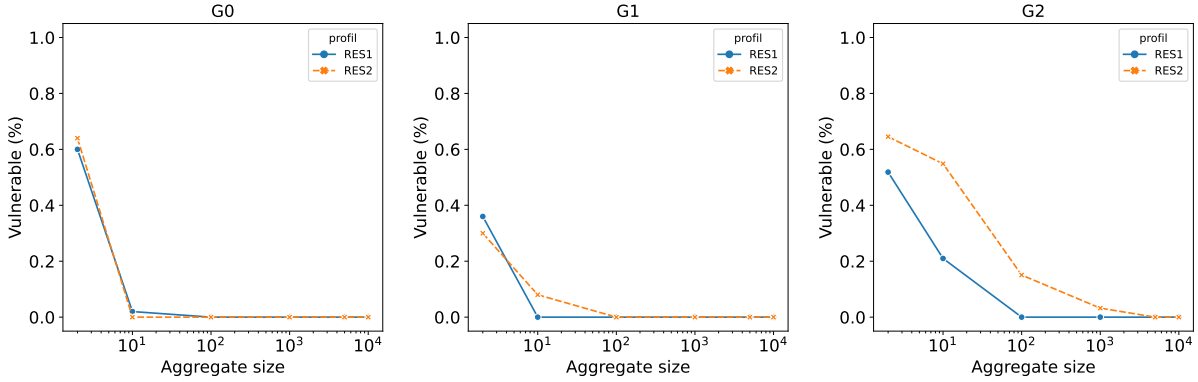
(a) Group G0 (b) Group G1 (c) Group G2

Figure 5.11 – STATS attack accuracy on the *ENEDIS* dataset per target and profile over a different period using the Subspace Alignment (SA) domain adaptation. $|\mathcal{S}^{\mathcal{A}}| = \{2, 100, 1000, 5000, 10000\}$, $|\mathcal{T}| = 1440$, training = 2021, testing = 2022.



(a) Group G0 (b) Group G1 (c) Group G2

Figure 5.12 – Percentage of vulnerable individuals on the *ENEDIS* dataset per target and profile over a different period using the CORAL domain adaptation. $|\mathcal{S}^{\mathcal{A}}| = \{2, 100, 1000, 5000, 10000\}$, $|\mathcal{T}| = 1440$, training = 2021, testing = 2022.



(a) Group G0 (b) Group G1 (c) Group G2

Figure 5.13 – Percentage of vulnerable individuals on the *ENEDIS* dataset per target and profile over a different period using the SA domain adaptation. $|\mathcal{S}^{\mathcal{A}}| = \{2, 100, 1000, 5000, 10000\}$, $|\mathcal{T}| = 1440$, training = 2021, testing = 2022.

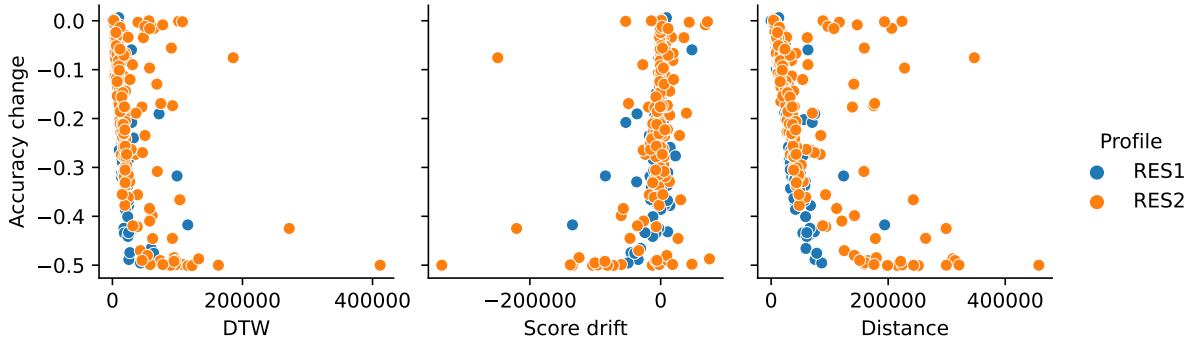
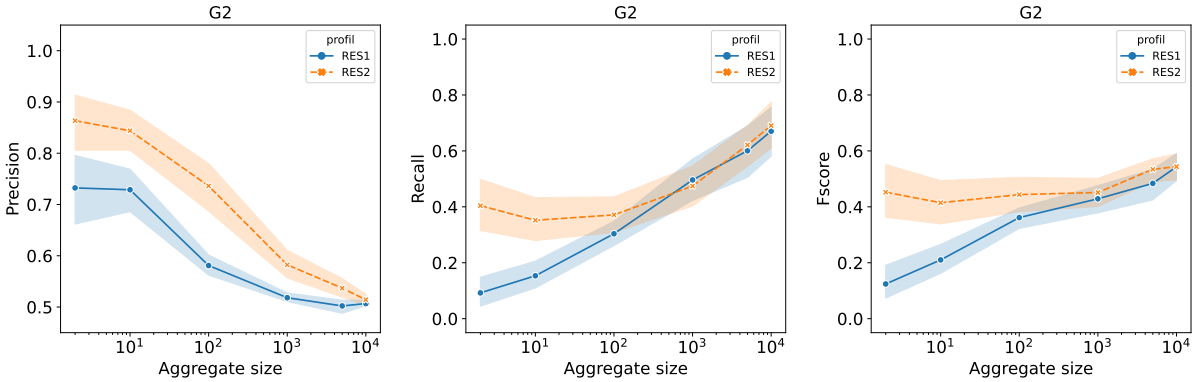


Figure 5.14 – Impact of the series changes between the same period attack and the different period on the accuracy. Series changes measured using using the DTW, the score drift and euclidian distance. Parameters: Enedis same period and different period (without domain adaptation), $|\mathcal{S}^{\mathcal{A}}| = 100$, $|\mathcal{T}| = 1440$, training = 2021, 2022, testing = 2022.



(a) Precision (b) Recall (c) Fscore

Figure 5.15 – STATS attack performance using the precision, recall and Fscore metrics on a different period. $|\mathcal{S}^{\mathcal{A}}| = \{2, 100, 1000, 5000, 10000\}$, $|\mathcal{T}| = 1440$, group = G2, training = 2021, testing = 2022.

mainly improves the recall (up to +50 %) rather than the precision (up to +10 %). Subsequently, the F-score increases with the recall.

5.6 Conclusion

This work introduces a technique to perform membership inference attacks on (threshold-based) aggregated time series. Given a known targeted series, an aggregate, and a dataset of similar series as the ones forming the aggregates, the adversary tries to re-identify the target's membership in the aggregate. We model the membership inference attack as a time series classification problem. This work presents a metric to estimate the vulnerability of series using the oddness score. We observe that the series with a higher score (relative to the other series) are more vulnerable to the attack. We perform experiments on two real-life datasets over multiple aggregate sizes, periods, and targets. Protecting every individual against the favorable attacker (i.e., performing the training and the attack on the same period) requires increasing the threshold. At most, we attack aggregates of 20,000 series, and 22 % of the RES2 G2 series remain vulnerable with a maximal accuracy of 0.77. However, only a few individuals remain vulnerable on large aggregates. For most series, we are not able to attack aggregates above 1000. Removing the most identifiable individuals from the aggregates allows the reduction of the aggregation threshold. We use domain adaptation methods to improve the attack's success in adversarial conditions. Domain adaptation methods significantly improve the attack results on small aggregates but do not improve the accuracy of large aggregates. Interesting future works include relaxing the background knowledge the attack requires (e.g., missing or approximate values). Besides, we focus on fast and state-of-the-art classification and domain adaptation methods due to the constraints of conducting large-scale experiments.

Chapter 6

Conclusion

As stated in the introduction, French legislation imposes the publication of electricity consumption time series. However, electricity consumption time series are personnal data. Therefore, the laws imposes the usage of threshold aggregation to protect the series. Besides, the legislation proposes to remove the "atypical" series before publication without defining "atypical". This work aims to understand the vulnerability of published electricity consumption time series. In particular, this thesis produces valuable information for choosing relevant aggregation thresholds. We also propose a metric to estimate the potential vulnerability of individual series prior to their aggregation. Section 6.1 summarizes our contributions. Section 6.2 extends our work by discussing the vulnerability of professional data. Section 6.3 proposes interesting future directions.

6.1 Summary of contributions

This thesis provides three main contributions. First, we perform a large-scale statistical study of French electricity consumption data. This study shows two main consumption patterns: a seasonal pattern with higher consumption during the winter and a daily pattern with higher consumption during the evening than at night. The consumption dispersion is linked to the mean and median consumption. The temperature strongly influences electricity consumption. Besides, the study shows that most individual series are unique, considering a few timestamps. Therefore, publishing pseudonymized individual series (i.e., without identifying information such as the name or contract number) leaves the series potentially identifiable and vulnerable to re-identification attacks. Degrading the data is insufficient to protect individual series while offering good utility.

We then propose two membership inference attacks. We adapt and improve existing concepts to cope with time series. All attacks are empirically tested against real-life electricity consumption data using large-scale and reproducible experiments. By doing so,

we manage to attack aggregates a magnitude larger than in the previous membership inference attacks in the literature.

First, the *SubSum* attack finds every series participating in an aggregate. It is modeled as a variant of the subset-sum problem. The attackers create one subset-sum constraint per timestamp, finding which series can form the aggregate on each timestamp. As the number of constraints increases, the set of individuals matching the constraints diminishes until the solver finds a single solution: the set of individuals participating in the aggregate. The attack requires knowing at least all the series in the aggregates. Our experiments show that the ability of the attacker to identify the members of an aggregate is related to the number of known individuals, the number of timestamps, and its available time. If the attacker has enough time, it is generally possible (in more than 90 % of the cases) to find all the individuals present in the aggregates if the number of published timestamps is greater than half the size of the attacker's known individuals. Note that the subset sum problem is an NP-hard problem. Therefore, it seems complicated to apply this attack against large populations. At most, we attack populations of 4,500 series requiring approximately a day of computation to attack a single aggregate. While slow, the computation time (i.e., a few days) remains acceptable while attacking aggregates containing a few thousand individuals.

Second, the *STATS* attack finds if a specific target participates in an aggregate. The attack is cast as a time series classification problem. This attack is fast (less than 1H) and requires limited knowledge from the attacker. It only requires knowing the targeted series at some point in time and a set of similar series as the ones in the aggregate. The attacker's known series can be on a different period than the one attacked. The attack can find an individual on large aggregates (i.e., thousands of individuals) while training the algorithm on the same period as the period attacked. At most, we successfully attack aggregates of 20,000 individuals. When it is carried out in adverse conditions (attack on a different period than the training period), the attack can no longer attack such important aggregates. It can identify up to 40 % of the targeted individuals for aggregates of less than 1,000 individuals. This attack is fast, requiring only a few minutes of calculation to attack a single aggregate. It is possible to improve the attack results using domain adaptation algorithms. The usage of domain adaptation improves the attack's results against small aggregates ($|\mathcal{S}^{\mathcal{A}}| < 1000$). Domain adaptation does not improve the attack's results on larger aggregates. We propose a metric estimating the potential vulnerability of a series

against this membership inference attack using the oddness score. Series with higher oddness scores have more chances to be identified by the STATS attack.

We answer our research question defined in the outline (Section 1.5): "how can a malicious actor breach the privacy of an aggregate, and what makes an aggregate safe from our attacker, considering the aggregate size and population?" First, this thesis shows that all the attacks require access to individual series. Therefore protecting and controlling access to individual series is crucial to prevent attacks on aggregates. Publishing pseudonymized individual series is risky as they are highly identifiable with minimal background knowledge. Second, while the *SubSum* attack requires significant time and computational power, the *STATS* attack can quickly run on a personal computer. The number of individuals in the aggregate and the series length influence the vulnerability. Long aggregates with few individuals are more vulnerable. This thesis shows that the current legal threshold is insufficient to protect everyone against our privacy attacks. Protecting every individual against the most knowledgeable attacker (i.e., performing training and testing in the same period) requires increasing the legal threshold. We do not make recommendations for a safe threshold able to protect everyone. At most, we attack aggregates of 20,000 series. Scaled to the general population, it could still represent thousands of individuals at risk. However, only a few individuals with a high oddness score remain vulnerable on large aggregates. Besides, the weaker attacker (i.e., performing the training and testing on a different period) yields significantly worse results on large aggregates. For most series, we are not able to attack aggregates above 1000. Removing the most identifiable individuals from the aggregates allows the reduction of the aggregation threshold. We confidently assume that it is possible to reduce the threshold to 1000 while protecting more than 90 % series against the most knowledgeable attacker (i.e., in the same period). Our oddness score could be used as a metric to estimate the vulnerability of each individual present in an aggregate. Removing the most identifiable individuals from the aggregates raises the question of data utility without introducing new vulnerabilities. For example, if we remove a single individual from an aggregate while keeping this individual in a higher level aggregate (e.g., publishing a city and an aggregate of cities), then subtracting the two aggregates reconstructs the consumption of the hidden individual. Note that the aggregate population strongly influences the attack results. It is necessary to analyze each publication individually to find any potential individuals vulnerable to a privacy attack.

6.2 Attacking professional data

This thesis focuses on personal data. However, smart meters also collect measurements from companies. As seen in Chapter 1, company measurements are not considered personal data under the privacy protection regulation. They can be published using a threshold of 3 series. However, they are still considered "commercially sensitive" and might represent a tempting target for industrial spies. The electric consumption of industrial sites reflects the site's physical location, the machinery used, and, generally, the company activity. Therefore, privacy protection methods and regulations should not be limited to personal data but should extend to any sensitive data requiring protection.

This section illustrates the vulnerability of professional electricity consumption time series by presenting our work with Diane Leblanc-Albarel at the EnergyDataHack 2022. The EnergyDataHack is a hackathon organized by the French Ministry of Defense [1]. The challenge aims to identify and cartograph French data centers using open-source intelligence. Our team got the best results at this hackathon. The full report is confidential. In this section, we only present a summary of the methodology and results. To accomplish this mission, we cross information publicly available from multiple sources: the Enedis open data, satellite imagery, and press articles. We locate the precise location of around 190 data centers, including 30 sensitive and secret ones.

To find the data centers, we use the following methodology. First, press articles provide the location and capacity of multiple data centers. Besides, some data centers were publicly referenced. Next, the energy consumption published on the Enedis open data highlights the approximate location of high-consumption entities. We filtered the aggregates probably related to data centers, research, and defense. The precise location can be found using the cartography of the electricity distribution network and satellite imagery. Enedis publishes the precise cartography of the electricity distribution network in open data. Data centers require a medium-voltage power supply. Buildings connected to the medium-voltage network are interesting guesses. Finally, data centers are easily identifiable from satellite imagery. As they consume a lot of energy, they produce heat. This heat is dispersed passively or using special cooling devices. Passively cooled buildings are identifiable by infrared satellite imagery as they are warmer than the surrounding buildings. Actively cooled buildings are cooled using very distinctive pieces of equipment easily distinguishable from the sky. France requires satellite imagery providers to blur sensitive

1. `https://energydatahack.challenkers.com/`

areas (including some data centers). It is possible to overcome this limitation by looking at overseas providers that do not blur the images.

6.3 Future works

A straightforward extension of the thesis would be the extension and improvement of the proposed attacks. First, it is possible to relax the background knowledge required by the attacks. Our attacks require time series without missing values. It would be interesting to cope with missing or approximated values within a time series. The *SubSum* attack requires all the series present in the aggregate. An interesting future work would be to cope with missing series. Our experiments provide insights into aggregate vulnerability while attacking specific populations. We perform the *STATS* attack using the Rocket classifier and compatible domain adaptation algorithms. We note the existence of deep classification models and domain adaptation methods. Such methods require extensive training time and computational power. It would be interesting to test other classifiers and domain adaptation algorithms trying to improve the attack's results. Another interesting future work would be the implementation of MIA as a service for publishers to estimate the vulnerability of their publications. Relaxing the attack prerequisites to consider an attacker crossing multiple consumption datasets (e.g., gas, water) and electricity consumption datasets is possible.

We focus on attacking Irish, Londoner, and French electricity consumption datasets. However, the attack results depend on the observed populations. It is necessary to study the vulnerability of each population individually. Our works focus on electricity consumption time series, but there is no doubt our algorithms can be applied to any time series. Our work may be used in any multivariate publications. Timestamp represents a new variable. In that case, however, the lack of correlation between the variables might require adapting the attack.

It is possible to work on new attacks by studying the aggregate vulnerability against new adversaries. For instance, we do not study attribute inference attacks due to technical limitations evoked in Section 2.1. We show that individual electricity consumption time series carries personal information about the household or the company activity. We do not study the impact of the aggregate on attribute inference: How many (and which) series must be aggregated to mask individual attributes? Without proving it, the ability to identify personal information is related to the ability to perform a membership attack.

If we cannot identify the individual, we cannot infer any attribute of this individual. Thus, we cannot tell whether a sensitive attribute (for example, the proportion of individuals with a particular religion) can be inferred from an aggregate.

This thesis does not study any defense mechanism besides threshold aggregation. We note that the individual or the publisher can implement several protection methods [Kal+10; RN10; Kal+11; ÁC11; KBB15; Aca+18] in addition to the aggregates. It would be interesting to challenge our attacks against other protection methods. Finally, we note the emergence of methods generating credible fake electricity consumption datasets (e.g., GAN) [YJS19; Fri+19; SOT22]. It would be interesting to evaluate the privacy of such methods.

BIBLIOGRAPHY

[09] *Directive 2009/72/EC of the european Parliament and of the Coundcil concerning common rules for the internal market in electricity and repealing Directive 2003/54/EC*, 2009.

[15] *Loi n°2015-992 du 17 août 2015 relative à la transition énergétique pour la croissance verte, Article 29*, 2015.

[16a] *Loi n° 2016-1321 du 7 octobre 2016 pour une République numérique*, 2016.

[16b] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*, 2016.

[17] *Mise à disposition du public de courbes de mesure relatives au transport et à la distribution d'électricité et de gaz naturel (Code de l'énergie, Articles D111-59 à D111-66)*, 2017.

[21] *Code de l'énergie, Article L322-8*, 2021.

[78] *Loi n° 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés*, 1978.

[ÁC11] Gergely Ács and Claude Castelluccia, « I Have a DREAM!(DiffeRentially privatE smArt Metering). », *in*: *Information hiding*, 2011.

[Aca+18] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti, « A Survey on Homomorphic Encryption Schemes: Theory and Implementation », *in*: *ACM Computing Surveys* (2018).

[ACF12] Joana M. Abreu, Francisco Câmara Pereira, and Paulo Ferrão, « Using pattern recognition to identify habitual behavior in residential electricity consumption », *in*: *Energy and Buildings* (2012).

[Apt03] Krzysztof Apt, *Principles of constraint programming*, Cambridge university press, 2003.

[AR13] Adrian Albert and Ram Rajagopal, « Smart Meter Driven Segmentation: What Your Consumption Says About You », *in*: *IEEE Transactions on Power Systems* (2013).

[Bag+16] A. Bagnall, Jason Lines, Aaron George Bostrom, James Large, and Eamonn J. Keogh, « The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances », *in*: *Data Mining and Knowledge Discovery* (2016).

[BB20] Luke A. Bauer and Vincent Bindschaedler, « Towards Realistic Membership Inferences: The Case of Survey Data », *in*: *Computer Security Applications Conference ACSAC*, 2020.

[Bir+12] Benjamin J. Birt, Guy R. Newsham, Ian Beausoleil-Morrison, Marianne M. Armstrong, Neil Saldanha, and Ian H. Rowlands, « Disaggregating categories of electrical energy end-use from whole-house hourly data », *in*: *Energy and Buildings* (2012).

[BSS13] Christian Beckel, Leyna Sadamori, and Silvia Santini, « Automatic Socio-Economic Classification of Households Using Electricity Consumption Data », *in*: *International Conference on Future Energy Systems*, 2013.

[Buc+13] Erik Buchmann, Klemens Böhm, Thorben Burghardt, and Stephan Kessler, « Re-identification of smart meter data », *in*: *Personal and ubiquitous computing* (2013).

[Büs+17] Niklas Büscher, Spyros Boukoros, Stefan Bauregger, and Stefan Katzenbeisser, « Two Is Not Enough: Privacy Assessment of Aggregation Schemes in Smart Metering », *in*: *Privacy Enhancing Technologies (PETS)* (2017).

[CER12] CER, *CER Smart Metering Project - Electricity Customer Behaviour Trial. Irish Social Science Data Archive. (Accessed May 11 2022)*, `https://www.ucd.ie/issda/data/commissionforenergyregulationcer`, 2012.

[Che+13] Dong Chen, Sean Barker, Adarsh Subbaswamy, David Irwin, and Prashant Shenoy, « Non-Intrusive Occupancy Monitoring Using Smart Meters », *in*: *ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, 2013.

[Che+22] Hanxiao Chen, Hongwei Li, Guishan Dong, Meng Hao, Guowen Xu, Xiaoming Huang, and Zhe Liu, « Practical Membership Inference Attack Against Collaborative Inference in Industrial IoT », *in*: *IEEE Transactions on Industrial Informatics* (2022).

[Chi+03] G. Chicco, R. Napoli, P. Postolache, M. Scutariu, and C. Toader, « Customer characterization options for improving the tariff offer », *in*: *IEEE Transactions on Power Systems* (2003).

[CN20] Aloni Cohen and Kobbi Nissim, « Linear Program Reconstruction in Practice », *in*: *Journal of Privacy and Confidentiality* (2020).

[Cra02] J. S. Cramer, « The Origins of Logistic Regression », *in*: *Econometrics eJournal* (2002).

[Cre+22] Ana-Maria Creţu, Federico Monti, Stefano Marrone, Xiaowen Dong, Michael Bronstein, and Yves-Alexandre de Montjoye, « Interaction data are identifiable even across long periods of time », *in*: *Nature Communications* (2022).

[De +13] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel, « Unique in the crowd: The privacy bounds of human mobility », *in*: *Scientific reports* (2013).

[DLS20] Aljoscha Dietrich, Dominik Leibenger, and Christoph Sorge, « On the Lack of Anonymity of Anonymized Smart Meter Data: An Empiric Study », *in*: *Conference on Local Computer Networks (LCN)*, 2020.

[DMT07] Cynthia Dwork, Frank McSherry, and Kunal Talwar, « The price of privacy and the limits of LP decoding », *in*: *ACM Symposium on Theory of Computing*, 2007.

[DN03] Irit Dinur and Kobbi Nissim, « Revealing Information While Preserving Privacy », *in*: *ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2003.

[DP16] Yves-Alexandre De Montjoye and Alex “Sandy” Pentland, « Response to Comment on “Unique in the shopping mall: On the reidentifiability of credit card metadata” », *in*: *Science* (2016).

[DPW20] Angus Dempster, François Petitjean, and Geoffrey I Webb, « ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels », *in*: *Data Mining and Knowledge Discovery* (2020).

[DSW21] Angus Dempster, Daniel F Schmidt, and Geoffrey I Webb, « Minirocket: A very fast (almost) deterministic transform for time series classification », *in*: *ACM SIGKDD*, 2021.

[Dwo+06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith, « Calibrating noise to sensitivity in private data analysis », *in*: *Theory of cryptography conference*, Springer, 2006.

[Dwo+17] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman, « Exposed! a survey of attacks on private data », *in*: *Annual Review of Statistics and Its Application* (2017).

[Dwo06] Cynthia Dwork, « Differential Privacy », *in*: *Automata, Languages and Programming, ICALP*, 2006.

[ENA15] Energy Networks Association (ENA), *Smart Meter Aggregation Assessment*, 2015.

[Fau+17] Anthony Faustine, Nerey Henry Mvungi, Shubi Felix Kaijage, and Michael Kisangiri, « A Survey on Non-Intrusive Load Monitoring Methodies and Techniques for Energy Disaggregation Problem », *in*: *CoRR* (2017).

[FEM17] Paul Francis, Sebastian Probst Eide, and Reinhard Munz, « Diffix: High-Utility Database Anonymization », *in*: *Privacy Technologies and Policy - Annual Privacy Forum, (APF)*, 2017.

[Fer+13] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars, « Unsupervised Visual Domain Adaptation Using Subspace Alignment », *in*: *2013 IEEE International Conference on Computer Vision* (2013).

[Fri+19] Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger, « Differentially Private Generative Adversarial Networks for Time Series, Continuous and Discrete Open Data », *in*: *ICT Systems Security and Privacy Protection*, 2019.

[GAM19] Simson Garfinkel, John M. Abowd, and Christian Martindale, « Understanding Database Reconstruction Attacks on Public Data », *in*: *Communications of the ACM* (2019).

[GH05] Marco Gruteser and Baik Hoh, « On the Anonymity of Periodic Location Samples », *in*: *Security in Pervasive Computing*, 2005.

[GJL12] Ulrich Greveler, Benjamin Justus, and Dennis Loehr, « Forensic content detection through power consumption », *in*: *IEEE International Conference on Communications (ICC)*, 2012.

[GKP13] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Nuñez del Prado Cortez, « De-anonymization Attack on Geolocated Data », *in*: *IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2013.

[GL16] Neil Zhenqiang Gong and Bin Liu, « You Are Who You Know and How You Behave: Attribute Inference Attacks via Users' Social Friends and Behaviors. », *in*: *USENIX Security Symposium*, 2016.

[Gol06] Philippe Golle, « Revisiting the Uniqueness of Simple Demographics in the US Population », *in*: *ACM Workshop on Privacy in Electronic Society (WEPS)*, 2006.

[HAP17] Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz, « Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning », *in*: *Conference on Computer and Communications Security, CCS*, 2017.

[Har92] George W. Hart, « Nonintrusive appliance load monitoring », *in*: *IEEE* (1992).

[Hay+17] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro, « LOGAN: Membership Inference Attacks Against Generative Models », *in*: *Privacy Enhancing Technologies (PETS)* (2017).

[HK70] Arthur E. Hoerl and Robert W. Kennard, « Ridge Regression: Biased Estimation for Nonorthogonal Problems », *in*: *Technometrics* (1970).

[Hom+08] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig, « Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays », *in*: *PLoS genetics* (2008).

[Hot36] Harold Hotelling, « Relations Between Two Sets of Variates », *in*: *Biometrika* (1936).

[Hu+22] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang, « Membership Inference Attacks on Machine Learning: A Survey », *in*: *ACM Computing Surveys* (2022).

[ISO18] ISO/IEC, *20889:2018 Privacy enhancing data de-identification terminology and classification of techniques*, 2018.

[Jab+15] Josef Jablonsky et al., « Benchmarks for current linear and mixed integer optimization solvers », *in*: *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis* (2015).

[Jay+21] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans, « Revisiting Membership Inference Under Realistic Assumptions », *in*: *Privacy Enhancing Technologies (PETS)* (2021).

[JJR11] Marek Jawurek, Martin Johns, and Konrad Rieck, « Smart Metering De-Pseudonymization », *in*: *Computer Security Applications Conference*, 2011.

[JJS17] Ming Jin, Ruoxi Jia, and Costas J. Spanos, « Virtual Occupancy Sensing: Using Smart Meters to Indicate Your Presence », *in*: *IEEE Transactions on Mobile Computing* (2017).

[Kal+10] Georgios Kalogridis, Costas Efthymiou, Stojan Z. Denic, Tim A. Lewis, and Rafael Cepeda, « Privacy for Smart Meters: Towards Undetectable Appliance Load Signatures », *in*: *IEEE International Conference on Smart Grid Communications*, 2010.

[Kal+11] G. Kalogridis, S.Z. Denic, T. Lewis, and R. Cepeda, « Privacy protection system and metrics for hiding electrical events », *in*: *International Journal of Security and Networks* (2011).

[KBB15] Stephan Kessler, Erik Buchmann, and Klemens Böhm, « Deploying and Evaluating Pufferfish Privacy for Smart Meter Data », *in*: *International Conference on Ubiquitous Intelligence and Computing (UIC-ATC-ScalCom)*, 2015.

[KPP04] Hans Kellerer, Ulrich Pferschy, and David Pisinger, « The Subset Sum Problem », *in*: *Knapsack Problems*, 2004.

[KRS13] Shiva Prasad Kasiviswanathan, Mark Rudelson, and Adam Smith, « The power of linear reconstruction attacks », *in*: *ACM-SIAM symposium on Discrete algorithms*, 2013.

[Lam22] Thoma Lamb, « Fixer les caractéristiques d'un instrument de politique publique controversé. Le compteur Linky entre économie d'énergie et économie de marché », *in*: *Revue française de science politique* (2022).

[LMW10] Mikhail A. Lisovich, Deirdre K. Mulligan, and Stephen B. Wicker, « Inferring Personal Information from Demand-Response Systems », *in*: *IEEE Security and Privacy* (2010).

[Lon+20] Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen, « A Pragmatic Approach to Membership Inferences on Machine Learning Models », *in*: *IEEE European Symposium on Security and Privacy (EuroS&P)* (2020).

[Mel+19] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov, « Exploiting Unintended Feature Leakage in Collaborative Learning », *in*: *IEEE Symposium on Security and Privacy (SP)*, 2019.

[MH20] Casimiro A. Curbelo Montañez and William Hurst, « A Machine Learning Approach for Detecting Unemployment Using the Smart Metering Infrastructure », *in*: *IEEE Access* (2020).

[Mid+21] Matthew Middlehurst, James Large, Michael Flynn, Jason Lines, Aaron George Bostrom, and A. Bagnall, « HIVE-COTE 2.0: a new meta ensemble for time series classification », *in*: *Machine Learning* (2021).

[Mol+10] Andrés Molina-Markham, Prashant Shenoy, Kevin Fu, Emmanuel Cecchet, and David Irwin, « Private Memoirs of a Smart Meter », *in*: *ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, 2010.

[Mon+15] Yves-Alexandre de Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex "Sandy" Pentland, « Unique in the shopping mall: On the reidentifiability of credit card metadata », *in*: *Science* (2015).

[MRT12] Eoghan McKenna, Ian Richardson, and Murray Thomson, « Smart meter data: Balancing consumer privacy concerns with legitimate applications », *in*: *Energy Policy* (2012).

[MSB23] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall, *Bake off redux: a review and experimental evaluation of recent time series classification algorithms*, 2023.

[MSS20] Santi Martıénez, Francesc Sebé, and Christoph Sorge, « Measuring privacy in smart metering anonymized data », *in*: *CoRR* (2020).

[Net13] UK Power Networks, *SmartMeter Energy Consumption Data in London Households (Accessed May 11 2022)*, `https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households`, 2013.

[NS08] Arvind Narayanan and Vitaly Shmatikov, « Robust De-anonymization of Large Sparse Datasets », *in*: *IEEE Symposium on Security and Privacy (SP)*, 2008.

[Pan+09] Sinno Jialin Pan, Ivor Wai-Hung Tsang, James Tin-Yau Kwok, and Qiang Yang, « Domain Adaptation via Transfer Component Analysis », *in*: *IEEE Transactions on Neural Networks* (2009).

[Phi17] Pierre Gastineau Philippe Vasset, *Armes de déstabilisation massive*, Fayard, 2017.

[Pod21] Emanuela Podda, « Shedding light on the legal approach to aggregate data under the GDPR and the FFDR », *in*: (2021).

[PTC17] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro, « What Does The Crowd Say About You? Evaluating Aggregation-based Location Privacy », *in*: *Privacy Enhancing Technologies (PETS)* (2017).

[PTC18] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro, « Knock Knock, Who's There? Membership Inference on Aggregate Location Data », *in*: *Network and Distributed System Security Symposium, NDSS*, 2018.

[PTC20] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro, « Measuring Membership Privacy on Aggregate Location Time-Series », *in*: *Measurement and Analysis of Computing Systems* (2020).

[Pyr19] Apostolos Pyrgelis, « Evaluating privacy-friendly mobility analytics on aggregate location data », PhD thesis, University of London, UK, 2019.

[Rah+18] Md.Atiqur Rahman, Tanzila Rahman, Robert Laganière, and Noman Mohammed, « Membership Inference Attack against Differentially Private Deep Learning Model », *in*: *Transactions on Data Privacy* (2018).

[Red+20] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani, « A survey on domain adaptation theory: learning bounds and theoretical guarantees », *in*: *arXiv: Learning* (2020).

[RLK21] Daniele Romanini, Sune Lehmann, and Mikko Kivelä, « Privacy and uniqueness of neighborhoods in social networks », *in*: *Scientific reports* (2021).

[RN10] Vibhor Rastogi and Suman Nath, « Differentially Private Aggregation of Distributed Time-Series with Transformation and Encryption », *in*: *ACM SIGMOD International Conference on Management of Data*, 2010.

[Rot+19] Cristina Rottondi, Marco Derboni, Dario Piga, and Andrea Emilio Rizzoli, « An optimisation-based energy disaggregation algorithm for low frequency smart meter data », *in*: *Energy Informatics* (2019).

[Sal+19] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes, « ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models », *in*: *Network and Distributed System Security Symposium, NDSS*, 2019.

[Sek+21] Vedran Sekara, Laura Alessandretti, Enys Mones, and Håkan Jonsson, « Temporal and cultural limits of privacy in smartphone app usage », *in*: *Scientific reports* (2021).

[SFS17] Baochen Sun, Jiashi Feng, and Kate Saenko, « Correlation Alignment for Unsupervised Domain Adaptation », *in*: *Domain Adaptation in Computer Vision Applications*, 2017.

[Sha48] Claude Elwood Shannon, « A mathematical theory of communication », *in*: *The Bell system technical journal* (1948).

[Sho+17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov, « Membership Inference Attacks Against Machine Learning Models », *in*: *IEEE Symposium on Security and Privacy (SP)*, 2017.

[SM21] Pascal Alexander Schirmer and Iosif Mporas, « On the non-intrusive extraction of residents' privacy-and security-sensitive information from energy smart meters », *in*: *Neural Computing and Applications* (2021).

[SOT22] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso, « Synthetic Data – Anonymisation Groundhog Day », *in*: *USENIX Security Symposium (USENIX Security 22)*, 2022.

[Swe00] Latanya Sweeney, « Simple demographics often identify people uniquely », *in*: *Health (San Francisco)* (2000).

[Swe02] Latanya Sweeney, « k-Anonymity: A Model for Protecting Privacy », *in*: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* (2002).

[Tru+21] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei, « Demystifying Membership Inference Attacks in Machine Learning as a Service », *in*: *IEEE Transactions on Services Computing* (2021).

[VKD13] Emmanouil Vogiatzis, Georgios Kalogridis, and Stojan Z. Denic, « Real-time and low cost energy disaggregation of coarse meter data », *in*: *IEEE PES Innovative Smart Grid Technologies Conference Europe*, 2013.

[Voy+22a] Antonin Voyez, Tristan Allard, Gildas Avoine, Pierre Cauchois, Elisa Fromont, and Matthieu Simonin, « Membership Inference Attacks on Aggregated Time Series with Linear Programming », *in*: *International Conference on Security and Cryptography (SECRYPT)*, July 2022.

[Voy+22b] Antonin Voyez, Tristan Allard, Gildas Avoine, Pierre Cauchois, Elisa Fromont, and Matthieu Simonin, *Unique in the Smart Grid -The Privacy Cost of Fine-Grained Electrical Consumption Data*, Nov. 2022.

[WD96] Leon Willenborg and Ton De Waal, *Statistical disclosure control in practice*, vol. 111, 1996.

[Wil45] Frank. Wilcoxon, « Individual Comparisons by Ranking Methods », *in*: *Biometrics* (1945).

[Wol20] Laurence A Wolsey, *Integer programming*, John Wiley & Sons, 2020.

[Yeo+18] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha, « Privacy risk in machine learning: Analyzing the connection to overfitting », *in*: *IEEE computer security foundations symposium (CSF)*, 2018.

[YJS19] Jinsung Yoon, James Jordon, and Mihaela van der Schaar, « PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees », *in*: *International Conference on Learning Representations*, 2019.

[Zha21] Youshan Zhang, « A Survey of Unsupervised Domain Adaptation for Visual Recognition », *in*: *ArXiv* (2021).

Appendix A

Glossary

Accuracy Classifier metric measuring the proportion of correct classifications.

Aggregate A sum or an average of multiple time series.

Attacker Malicious actor trying to infer private information.

Client An individual (household) or any entity (i.e., a com- pany) consuming electricity and equipped with a meter.

Confusion matrix the proportion of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) examples classified by the model.

Constraint A logical relation between variables.

Consumption A consumption or power measurement (in Watt -W- or Watt per hour -Wh-).

DSO **D**istribution **S**ystem **O**perator. DSOs are in charge of the electricity distribution in France.

Fscore Metric evaluating the model capacity to perform correct positive classifications.

Ground truth Labeled series with true information.

Linky The Enedis smart meter.

Meter A device measuring the electricity consumed by a household and reporting it to a central server.

Measurement A point measured by the meter at a timestamp. The measurements are at a 1/2h step and to the W. All measurements are between [0; 36000]W.

Membership Inference Attack (MIA) Privacy attack aiming to find is a series is present in a publication.

NILM **N**on **I**ntrusive **L**oad **M**onitoring. A field aimed at the extraction of information from electricity consumption time series.

Oddness score Score indicating how far a series is from the mean series.

Overfit A model is said to overfit when it remembers the training values instead of learning a trend.

Perimeter Set of meters within a dataset (or a subset). A perimeter is represented with the following tuple : (Geographical area; profiles) (example: (NAT; RES1) for the perimeter containing all the RES1 meters in the National geographical perimeter)

Profile Contract type associated with a meter and a way of consuming electricity.

RES Residential meters.

RES1 BASE meters (same price every time) $\leq$ 6 kVA

RES2 HPHC meters (cheaper during certain hours, here undefined)

RES11 BASE meters (same price every time) $>$ 6 kVA

RESAutre mixing of all other RES profiles.

RES* RES1 $\cup$ RES2 $\cup$ RES11 $\cup$ RESAutre

PRO Professional meters.

PRO1 BASE meters (same price every time)

PRO2 HPHC meters (cheaper during certain, here undefined, hours)

PRO5 Public lighting.

PROAutre Mixing of all other PRO profiles.

PRO* PRO1 $\cup$ PRO2 $\cup$ PRO5 $\cup$ PROAutre

RES*+PRO* RES* $\cup$ PRO*

Geographical area A geographical area. Either the whole metropolitan France or an administrative department.

NAT Contains all of metropolitan France.

13 The Mediterranean department "Bouche-du-Rhône" (13).

29 The oceanic department "Finistère" (29).

51 The continental department "Marne" (51).

75 The alpine department of "Haute-Savoie" (74).

75 The department "Paris" (75).

Precision The precision as the proportion of correct positive classifications.

Pseudonymization Protection method consisting of removing any personal information from the data.

Publisher Entity publishing aggregates.

Recall The proportion of positive labels correctly identified by the model.

Solution Set of individual matching a set of constraints.

Threshold Minimum number of series to put in the aggregate (legal).

Time Series a sequence of timestamped measurements.

Unique A series is unique if it is the only one with a set of values.

Uniqueness Proportion of unique series.

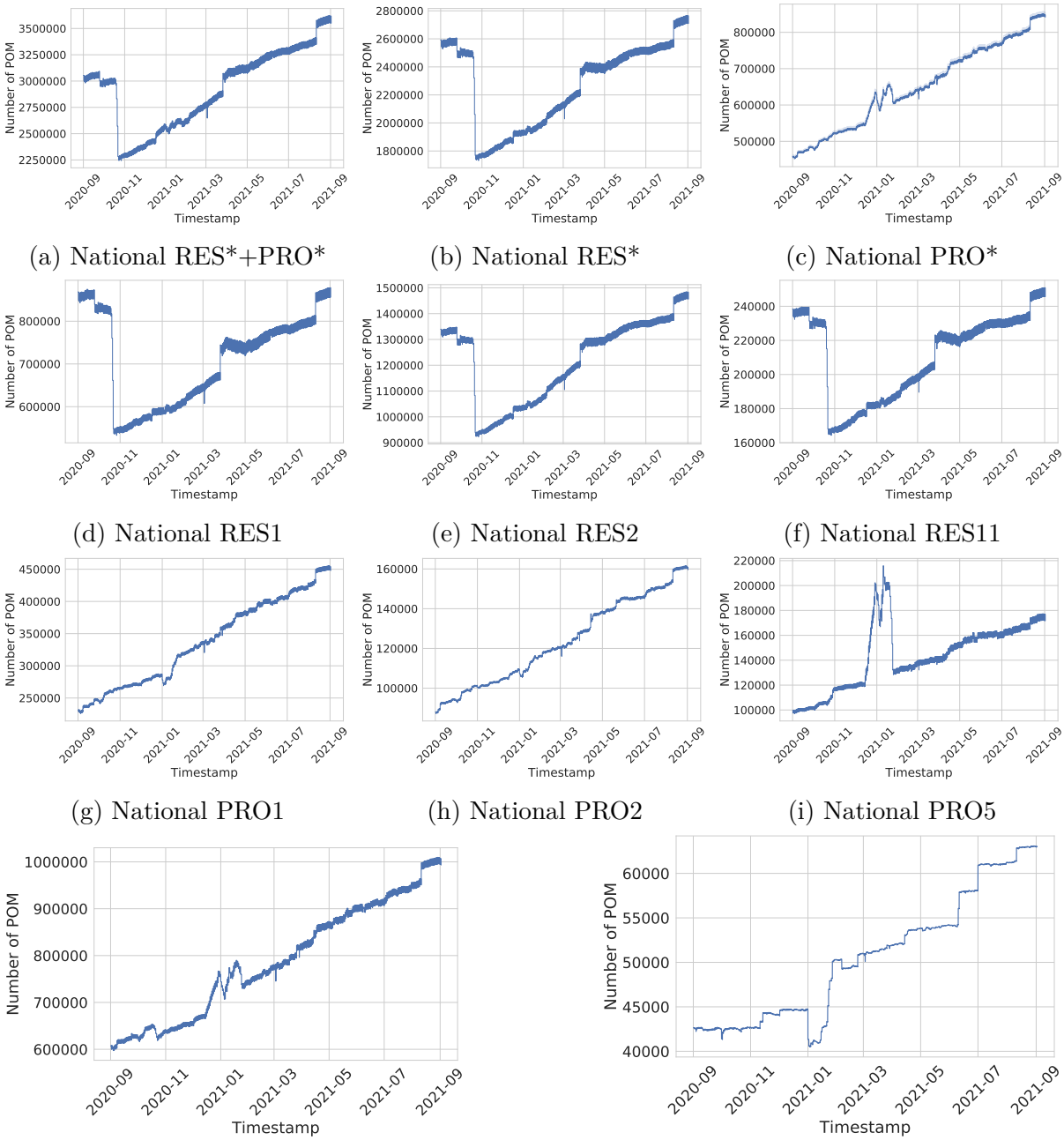
Variable unknown value within a definition domain.

Shadow model MIA methodology consisting of training aan attack model on fake publications.

Appendix B

Enedis data statistical study: all figures

All of the individual figures generated for the study in Section 3.4, per perimeter and analysis without comments.



(a) National RES*+PRO*

(b) National RES*

(c) National PRO*

(d) National RES1

(e) National RES2

(f) National RES11

(g) National PRO1

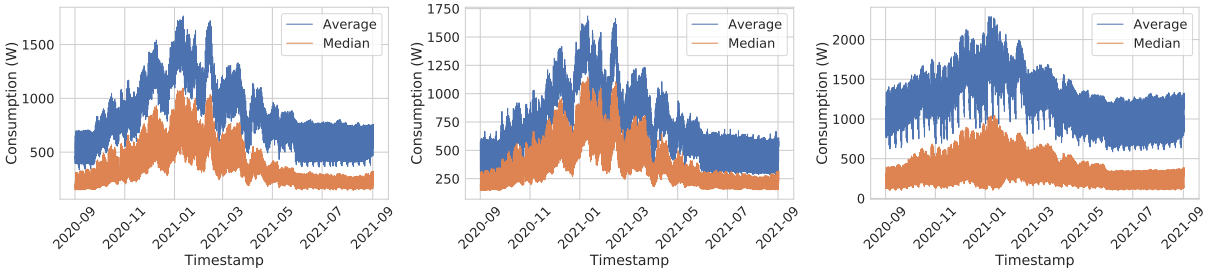
(h) National PRO2

(i) National PRO5

(j) National RESAutre

(k) National PROAutre

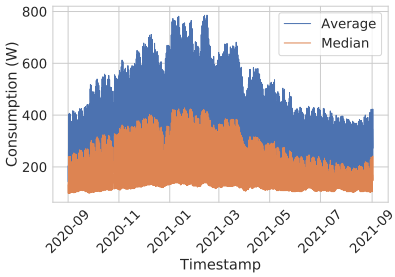
Figure B.1 – Number of POM per 1/2h



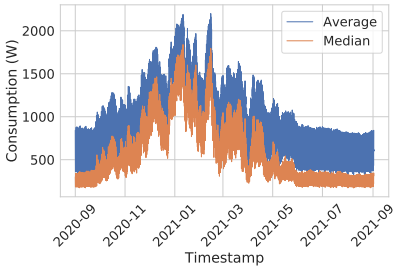
(a) National RES*+PRO*

(b) National RES*

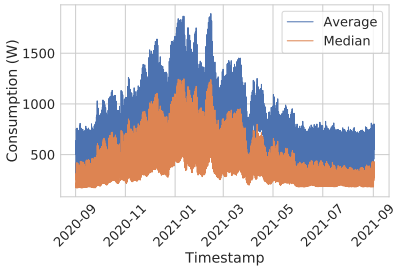
(c) National PRO*



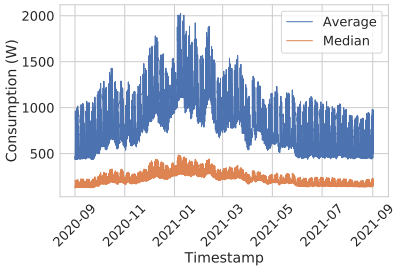
(d) National RES1



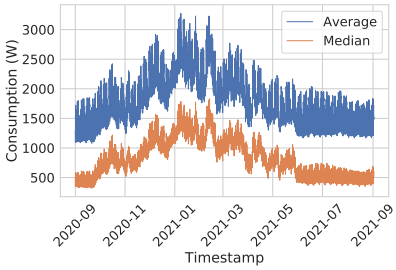
(e) National RES2



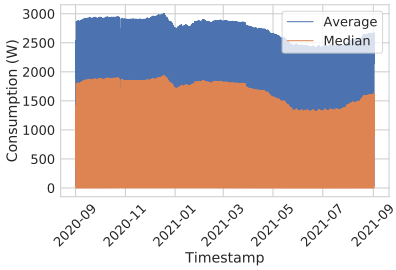
(f) National RES11



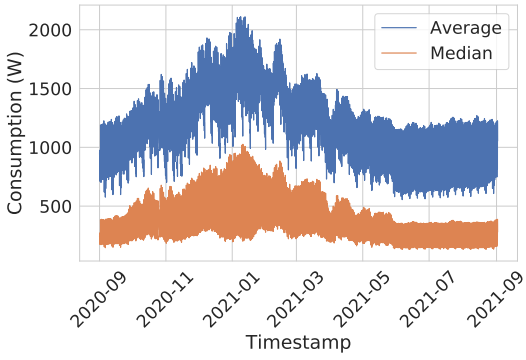
(g) National PRO1



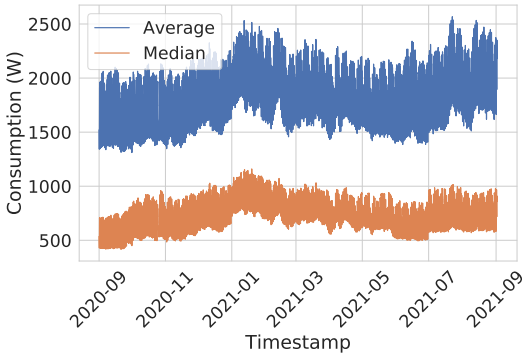
(h) National PRO2



(i) National PRO5

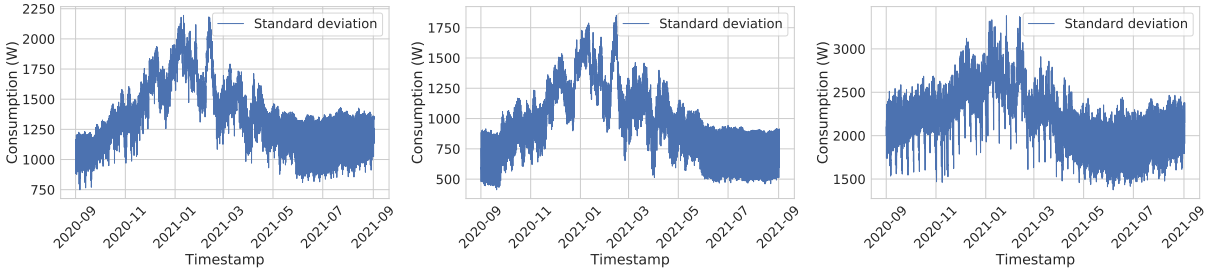


(j) National RESAutre

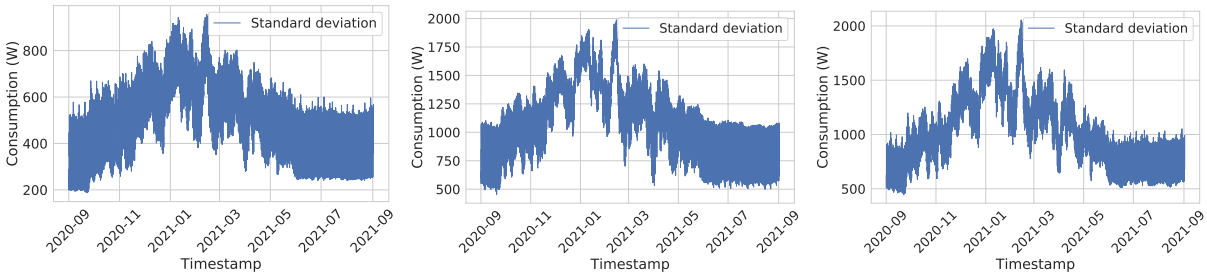


(k) National PROAutre

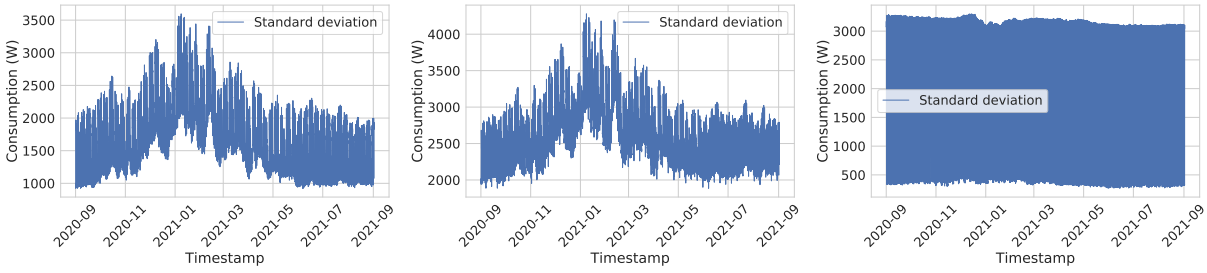
Figure B.2 – Average and median per 1/2h



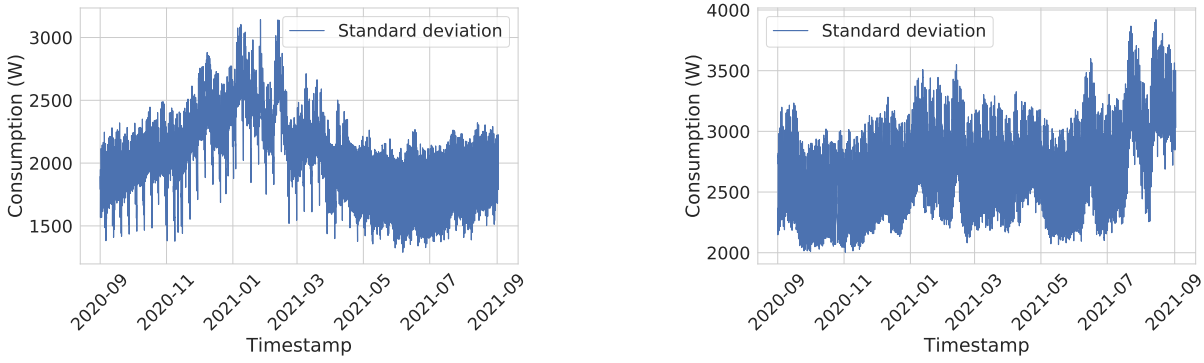
(a) National RES*+PRO* (b) National RES* (c) National PRO*



(d) National RES1 (e) National RES2 (f) National RES11

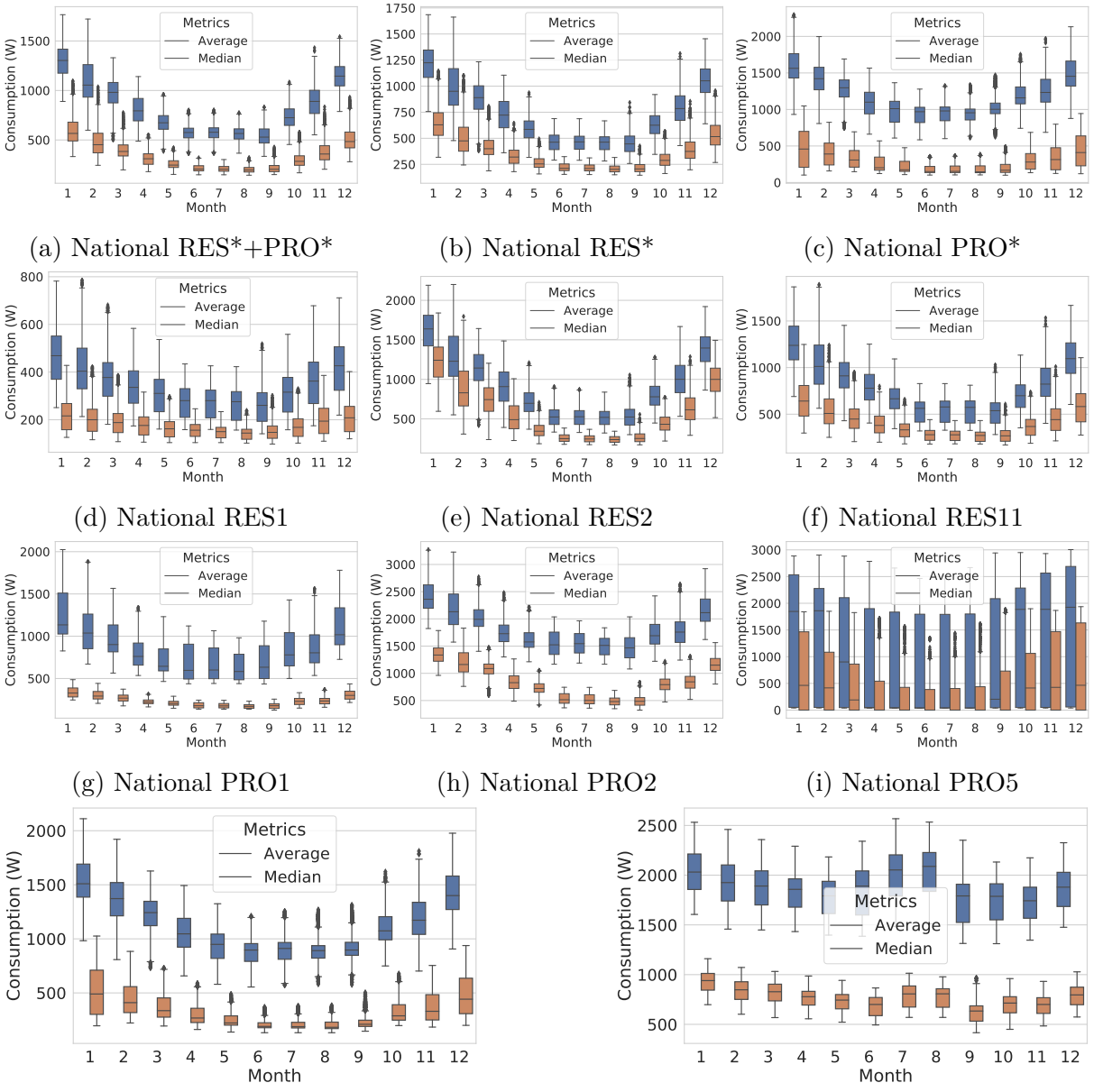


(g) National PRO1 (h) National PRO2 (i) National PRO5



(j) National RESAutre (k) National PROAutre

Figure B.3 – Standard deviation per 1/2h



(a) National RES*+PRO*

(b) National RES*

(c) National PRO*

(d) National RES1

(e) National RES2

(f) National RES11

(g) National PRO1

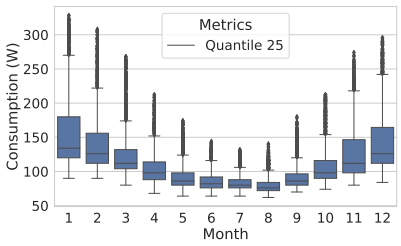
(h) National PRO2

(i) National PRO5

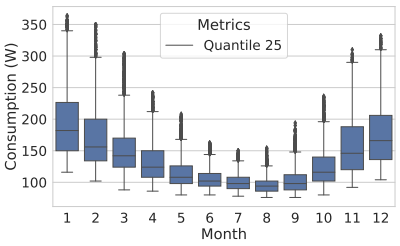
(j) National RESAutre

(k) National PROAutre

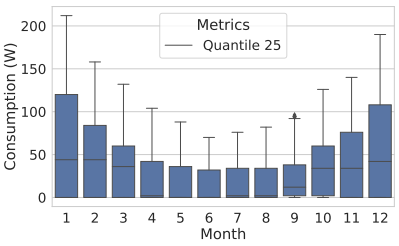
Figure B.4 – Average and median by month



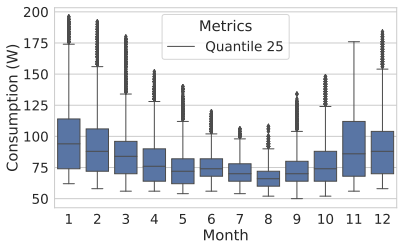
(a) National RES*+PRO*



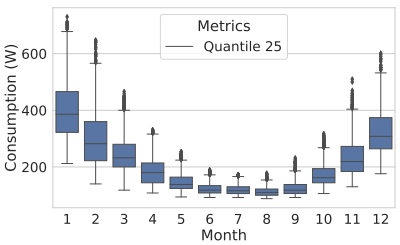
(b) National RES*



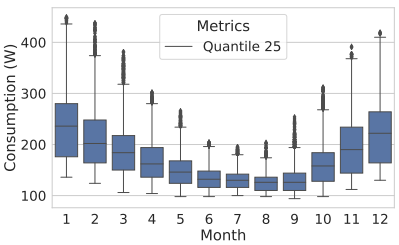
(c) National PRO*



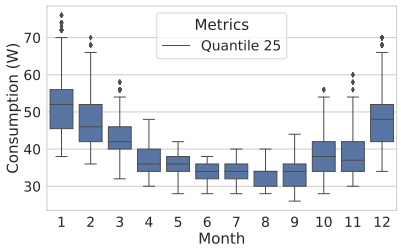
(d) National RES1



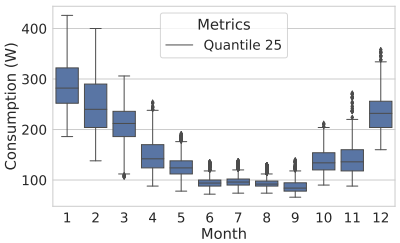
(e) National RES2



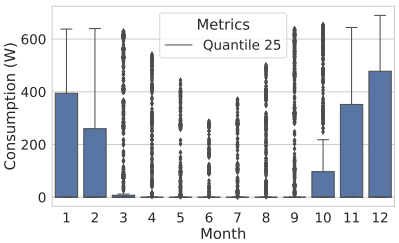
(f) National RES11



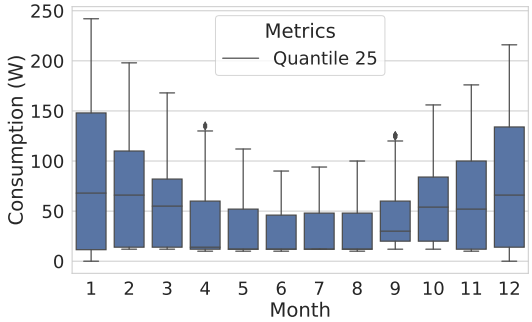
(g) National PRO1



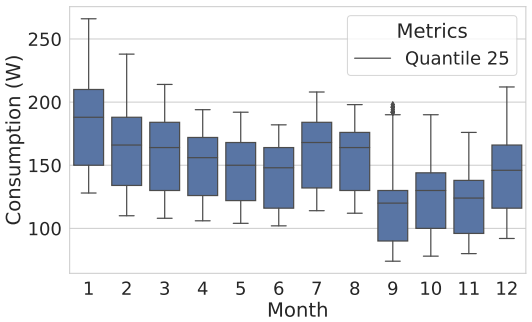
(h) National PRO2



(i) National PRO5

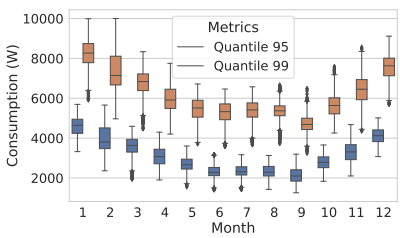


(j) National RESAutre

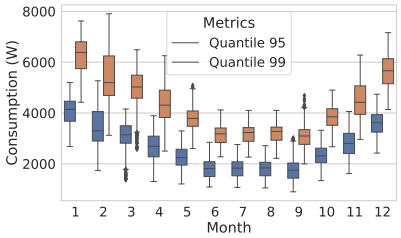


(k) National PROAutre

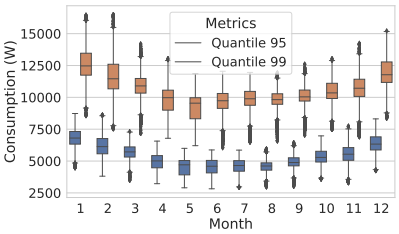
Figure B.5 – Q25 by month



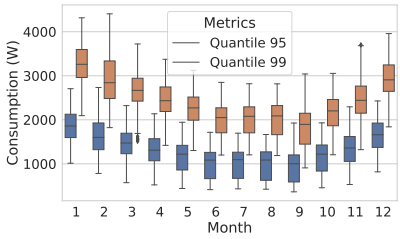
(a) National RES*+PRO*



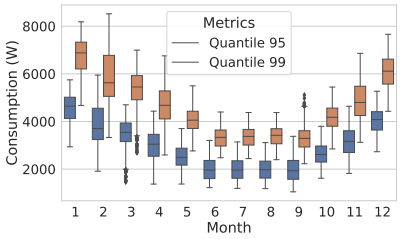
(b) National RES*



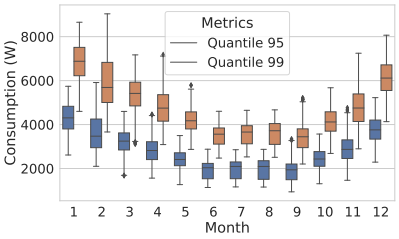
(c) National PRO*



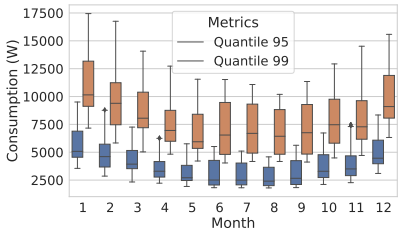
(d) National RES1



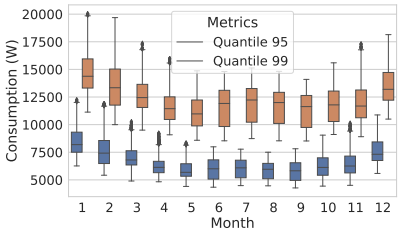
(e) National RES2



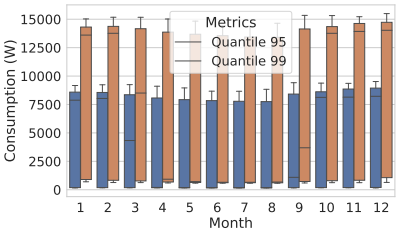
(f) National RES11



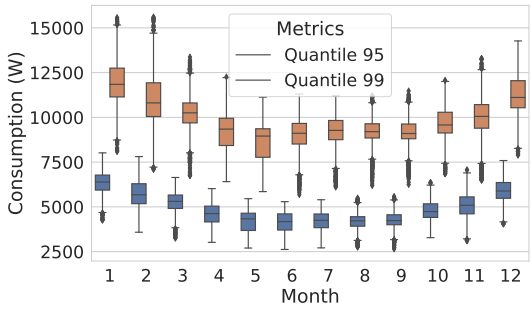
(g) National PRO1



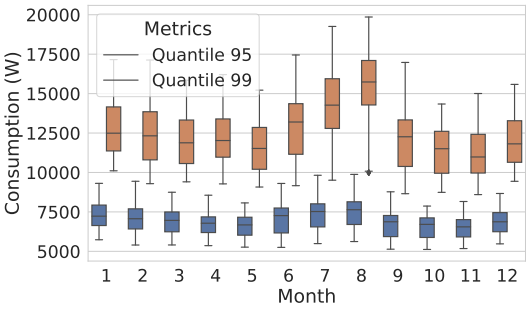
(h) National PRO2



(i) National PRO5

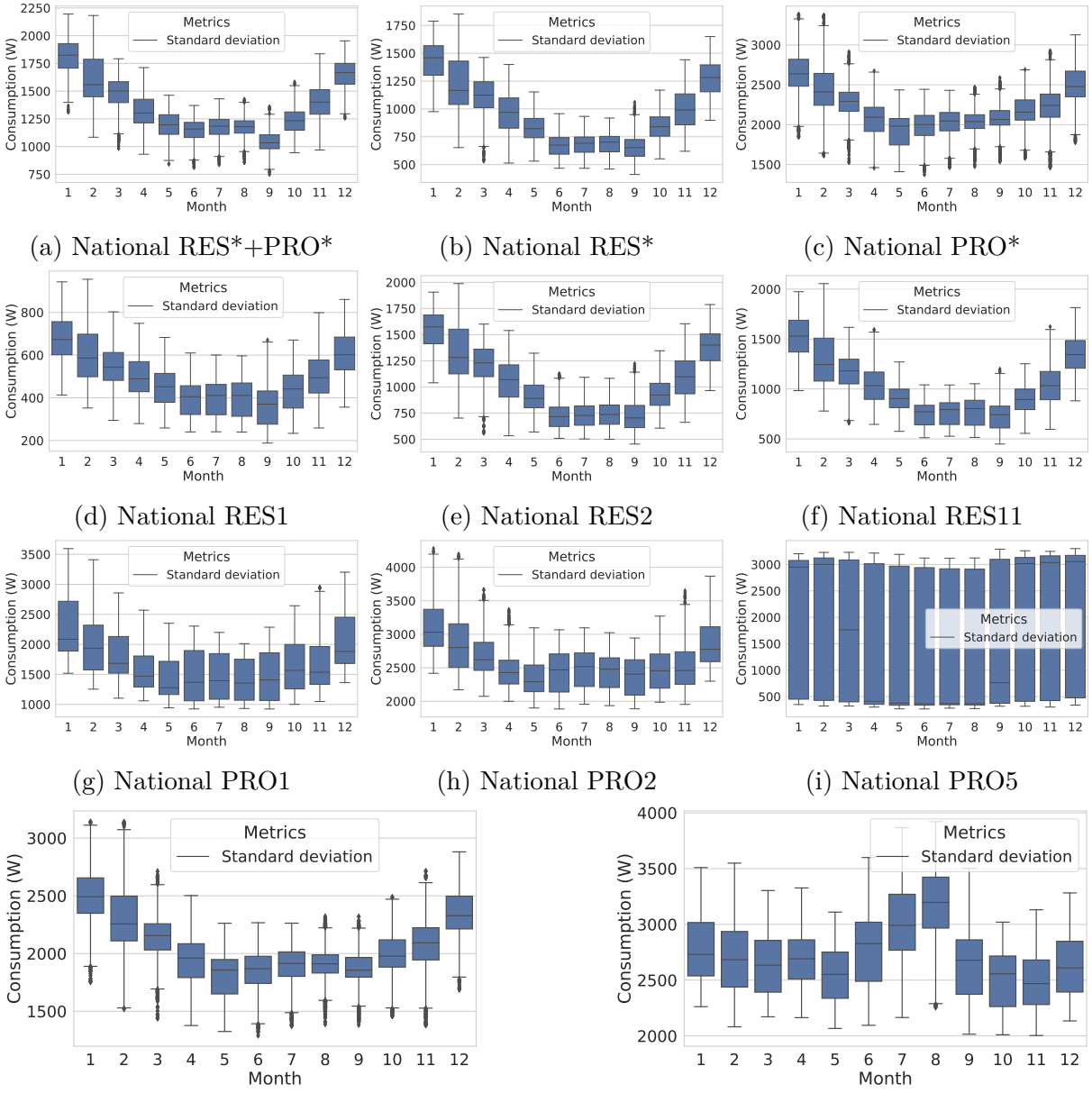


(j) National RESAutre



(k) National PROAutre

Figure B.6 – Q95 and Q99 by month



(a) National RES*+PRO*

(b) National RES*

(c) National PRO*

(d) National RES1

(e) National RES2

(f) National RES11

(g) National PRO1

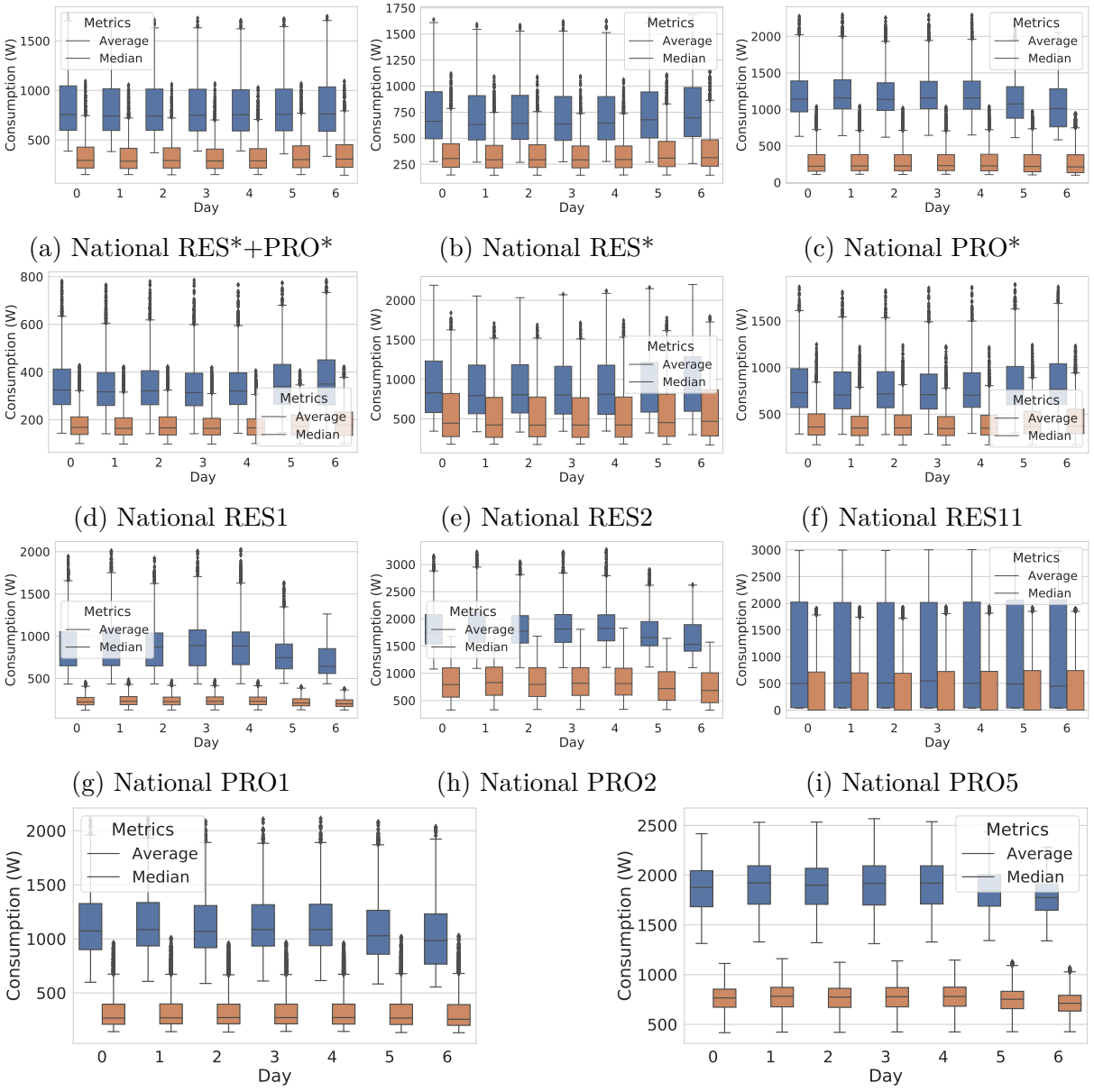
(h) National PRO2

(i) National PRO5

(j) National RESAutre

(k) National PROAutre

Figure B.7 – Standard deviation by month



(a) National RES*+PRO*

(b) National RES*

(c) National PRO*

(d) National RES1

(e) National RES2

(f) National RES11

(g) National PRO1

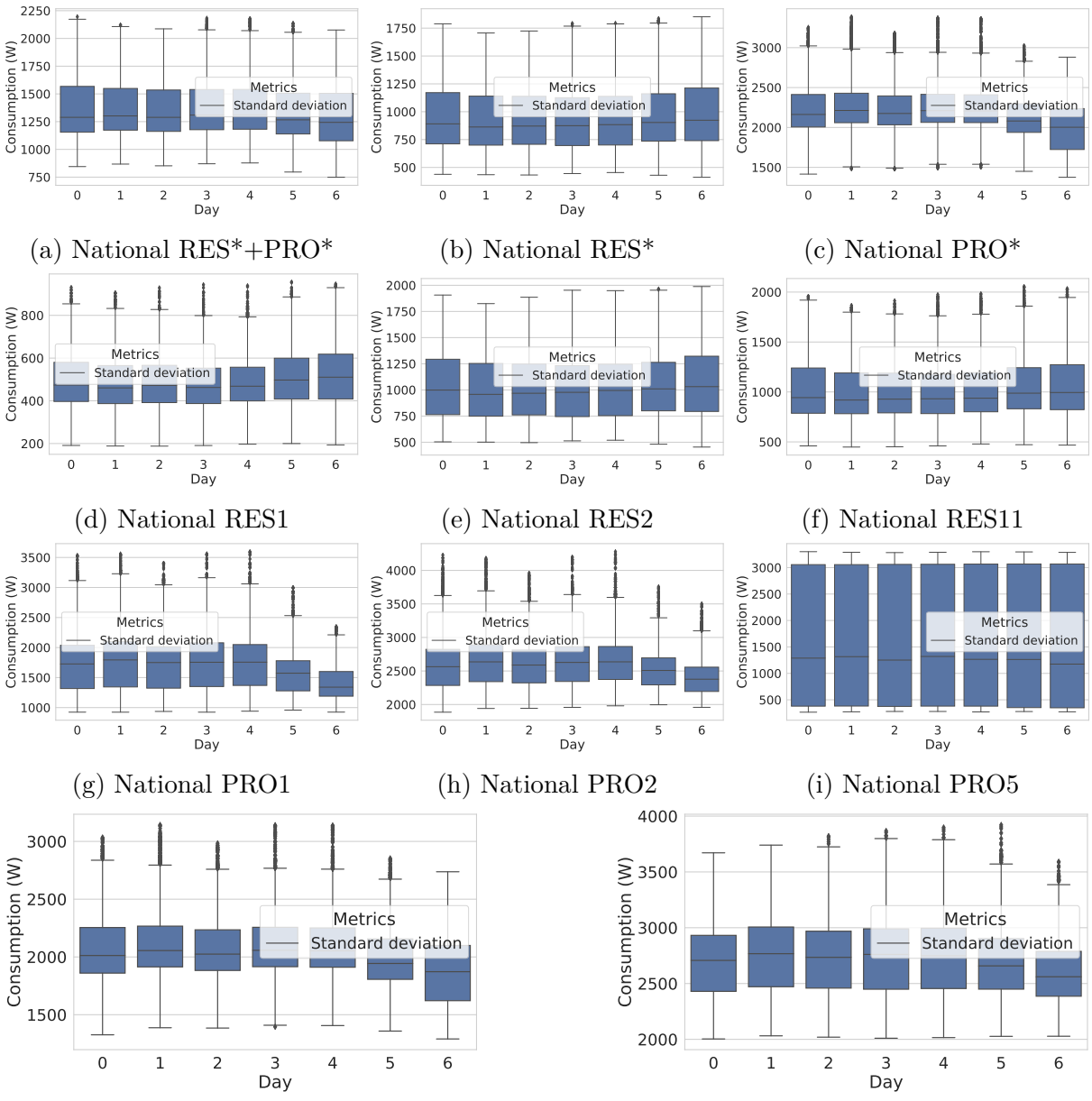
(h) National PRO2

(i) National PRO5

(j) National RESAutre

(k) National PROAutre

Figure B.8 – Average and median per day



(a) National RES*+PRO*

(b) National RES*

(c) National PRO*

(d) National RES1

(e) National RES2

(f) National RES11

(g) National PRO1

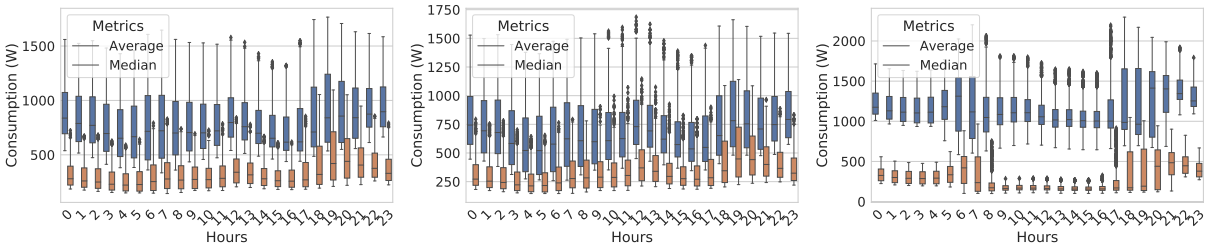
(h) National PRO2

(i) National PRO5

(j) National RESAutre

(k) National PROAutre

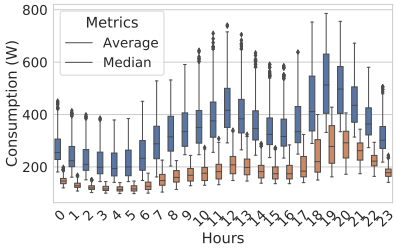
Figure B.9 – Standard deviation per days



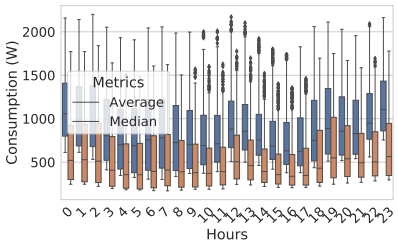
(a) National RES*+PRO*

(b) National RES*

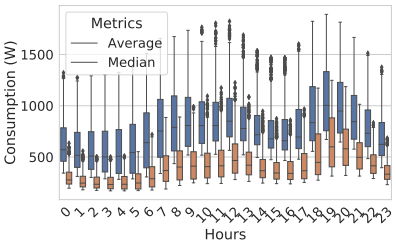
(c) National PRO*



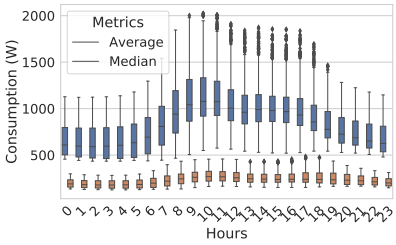
(d) National RES1



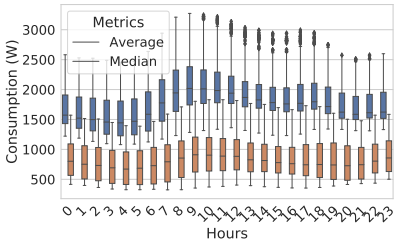
(e) National RES2



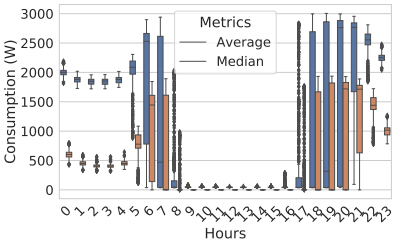
(f) National RES11



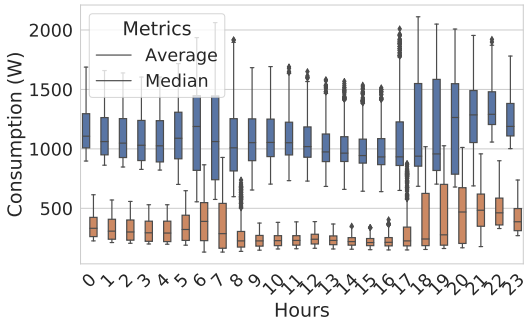
(g) National PRO1



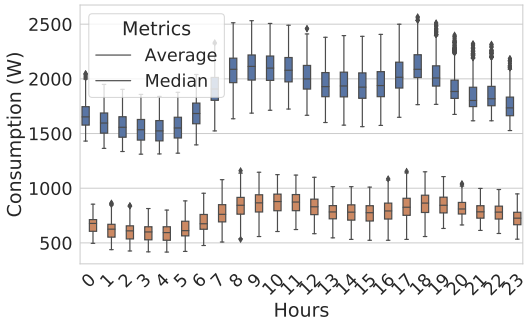
(h) National PRO2



(i) National PRO5

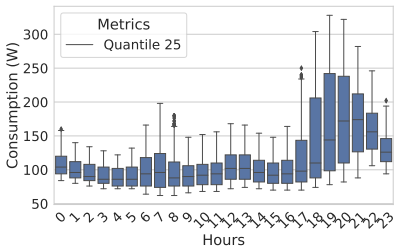


(j) National RESAutre

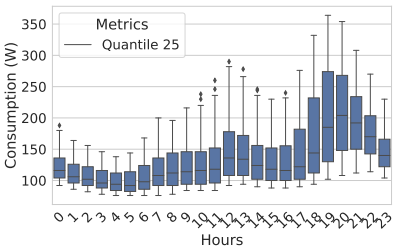


(k) National PROAutre

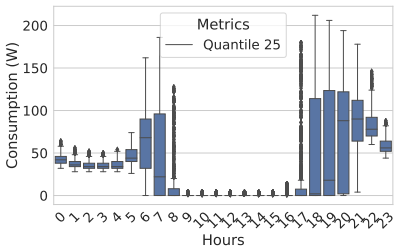
Figure B.10 – Average and median per hours



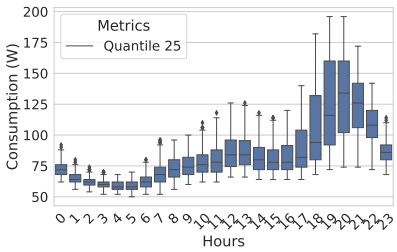
(a) National RES*+PRO*



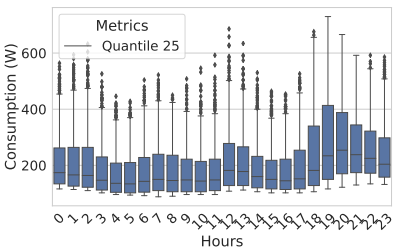
(b) National RES*



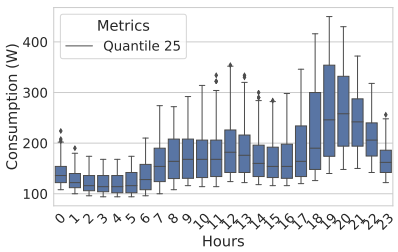
(c) National PRO*



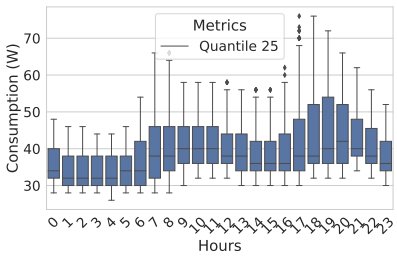
(d) National RES1



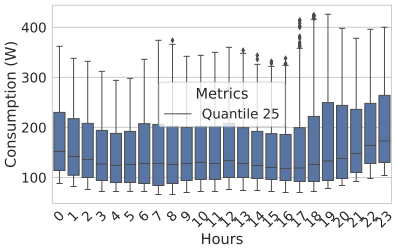
(e) National RES2



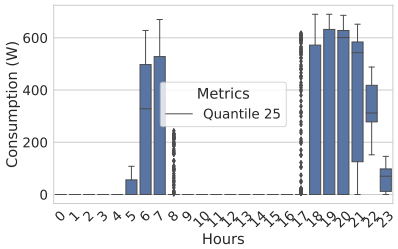
(f) National RES11



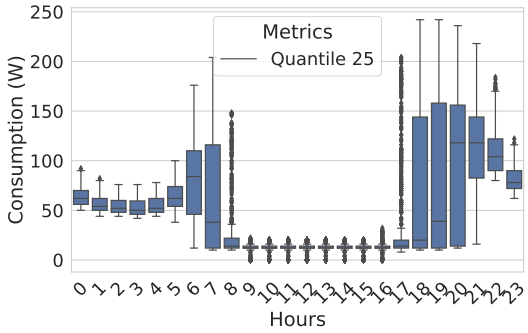
(g) National PRO1



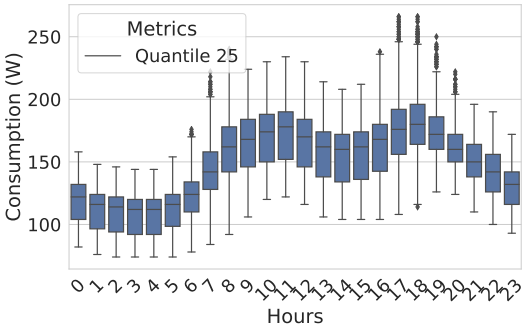
(h) National PRO2



(i) National PRO5

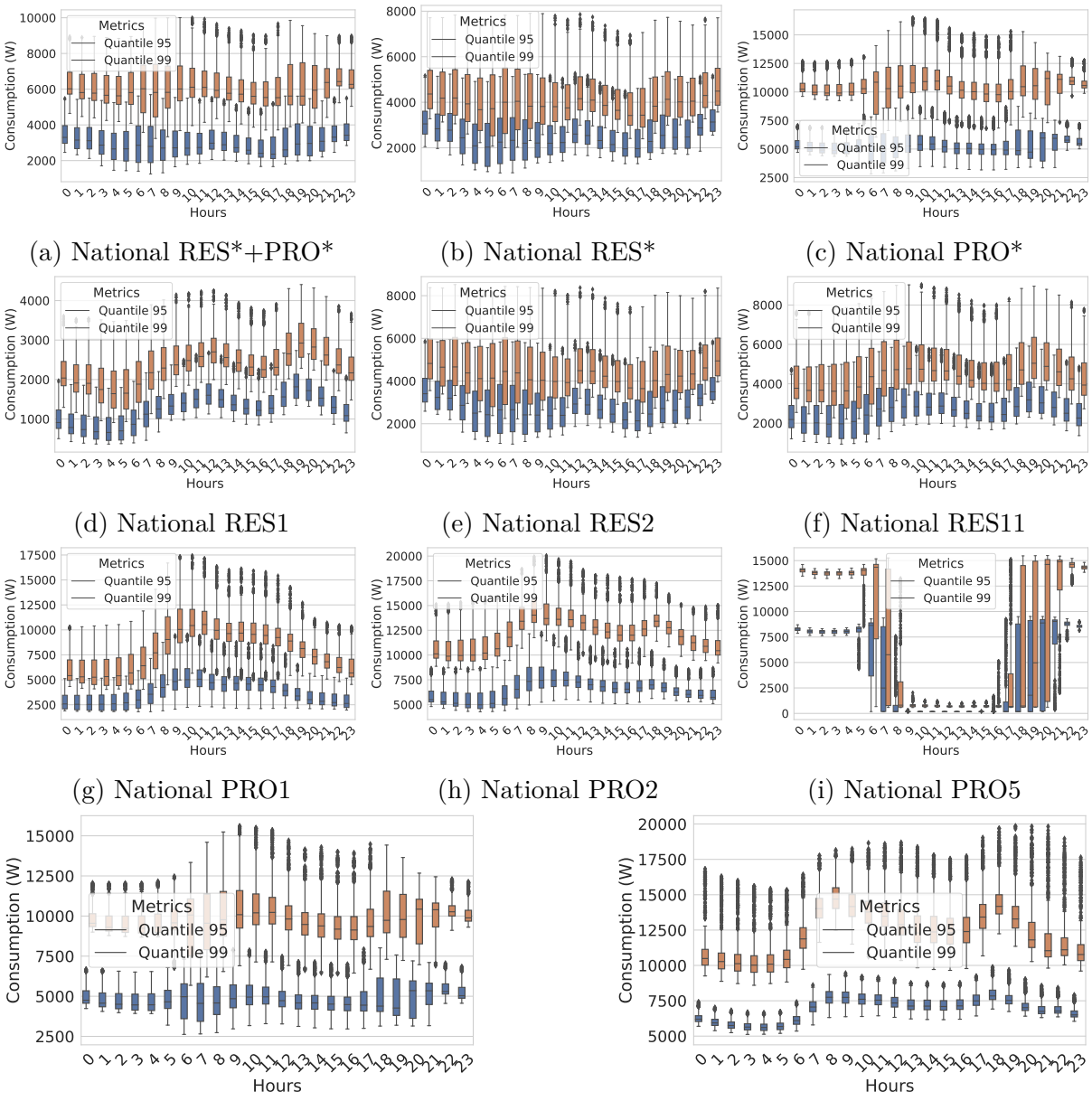


(j) National RESAutre



(k) National PROAutre

Figure B.11 – Quartile 25 per hours



(a) National RES*+PRO*

(b) National RES*

(c) National PRO*

(d) National RES1

(e) National RES2

(f) National RES11

(g) National PRO1

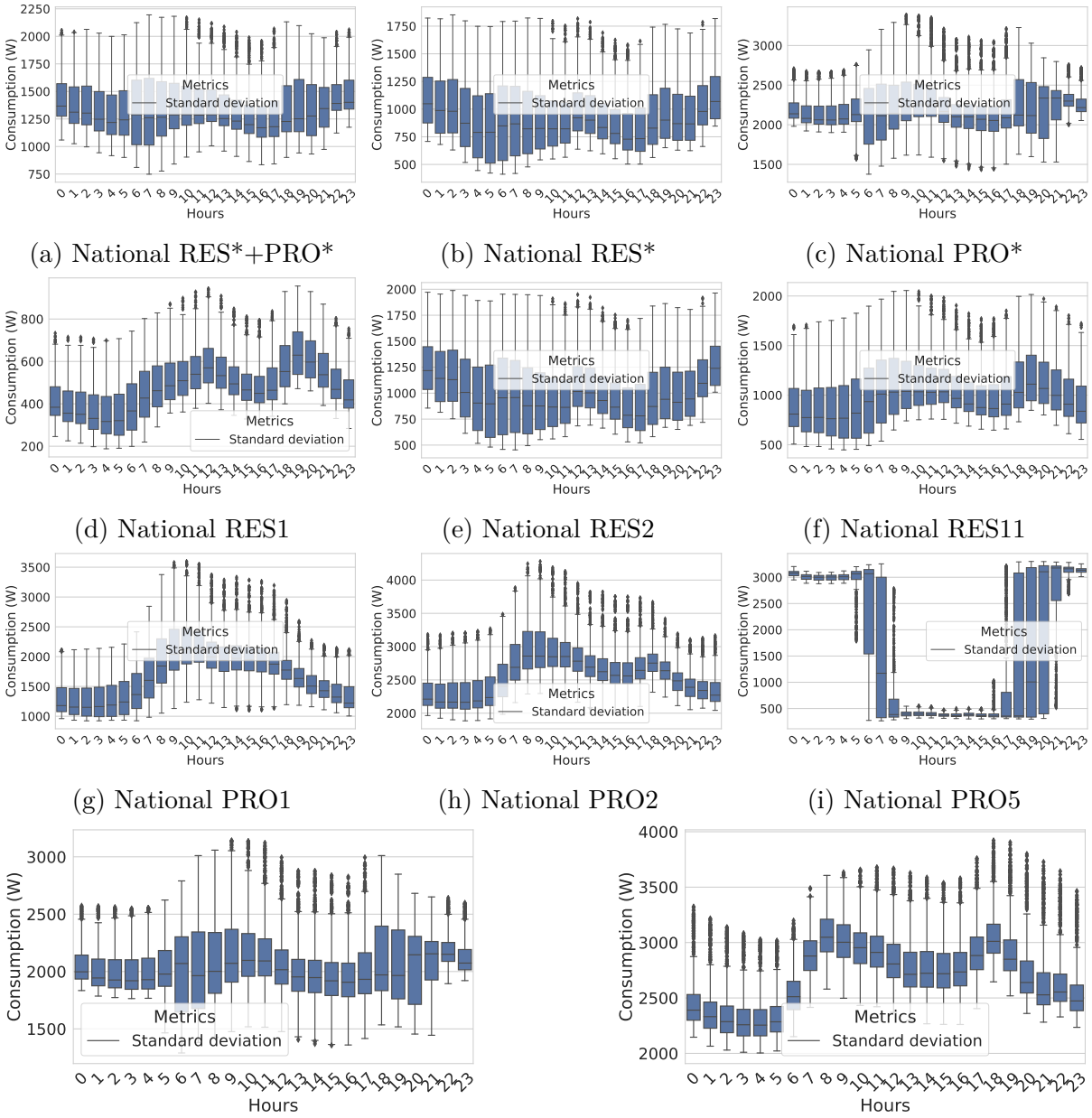
(h) National PRO2

(i) National PRO5

(j) National RESAutre

(k) National PROAutre

Figure B.12 – Quartile 95 and 99 per hours



(a) National RES*+PRO*

(b) National RES*

(c) National PRO*

(d) National RES1

(e) National RES2

(f) National RES11

(g) National PRO1

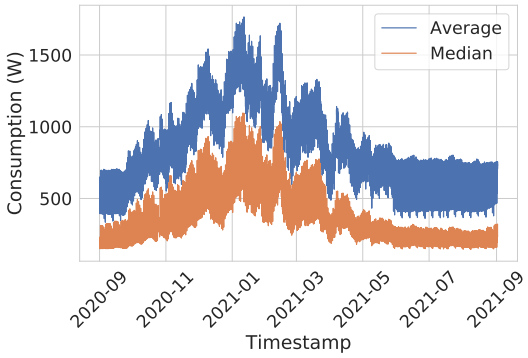
(h) National PRO2

(i) National PRO5

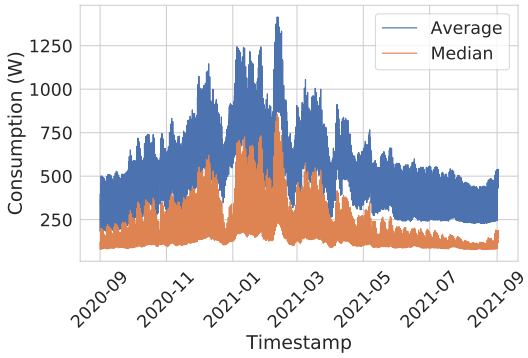
(j) National RESAutre

(k) National PROAutre

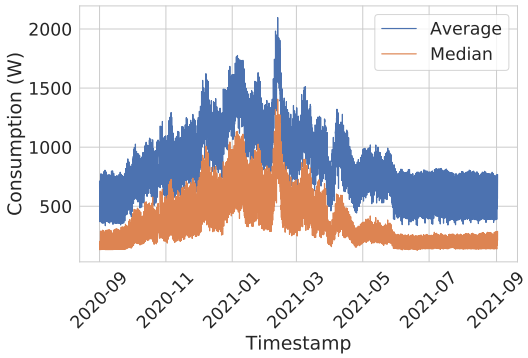
Figure B.13 – Standard deviation per hours



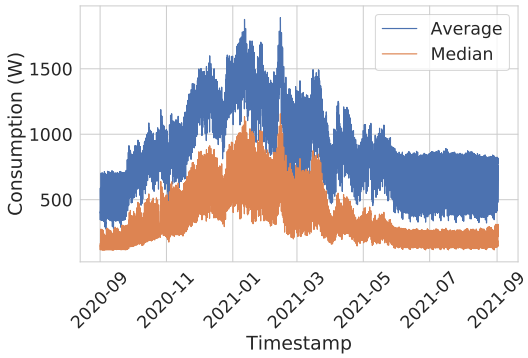
(a) National RES*+PRO*



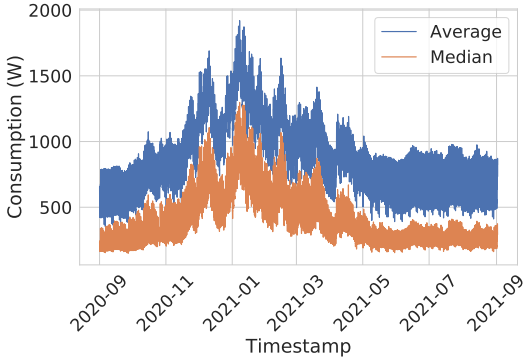
(b) 75 RES*+PRO*



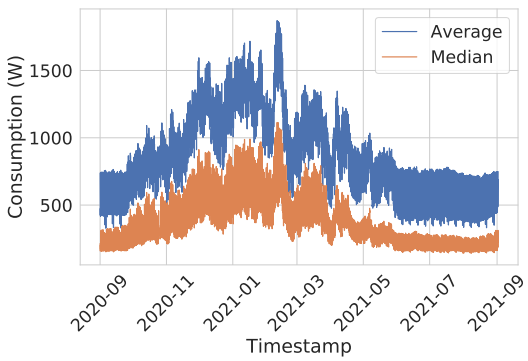
(c) 29 RES*+PRO*



(d) 74 RES*+PRO*

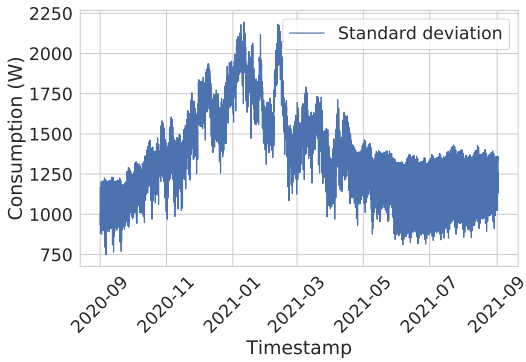


(e) 13 RES*+PRO*

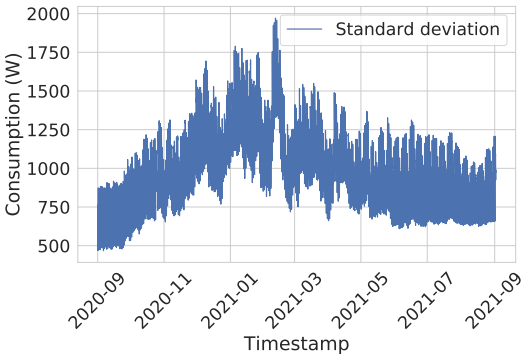


(f) 51 RES*+PRO*

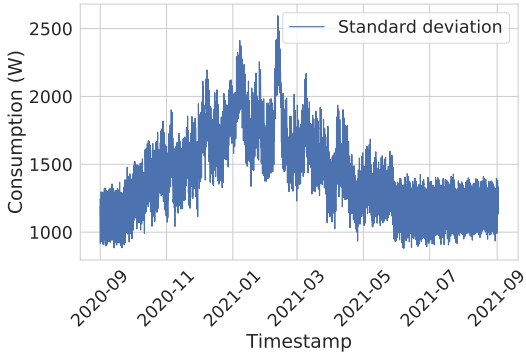
Figure B.14 – Average and median per 1/2h at regional levels



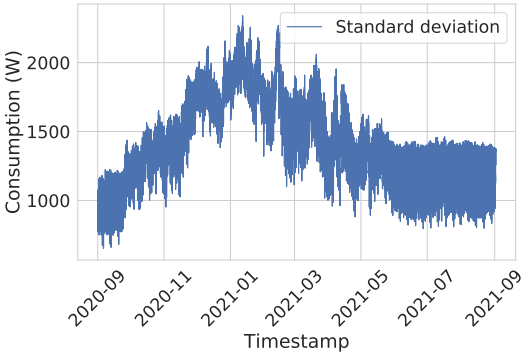
(a) National RES*+PRO*



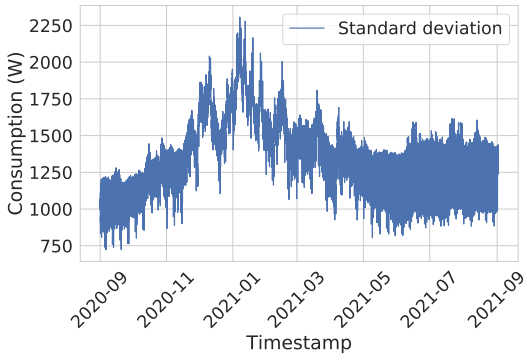
(b) 75 RES*+PRO*



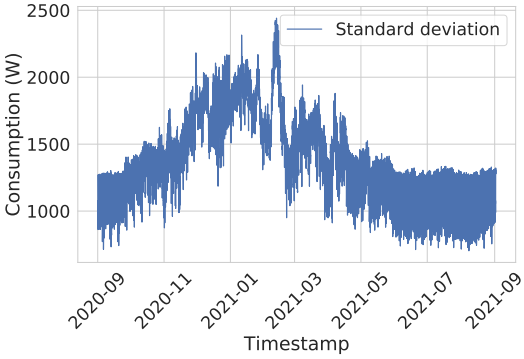
(c) 29 RES*+PRO*



(d) 74 RES*+PRO*

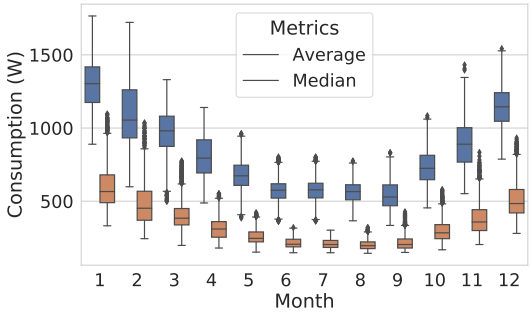


(e) 13 RES*+PRO*

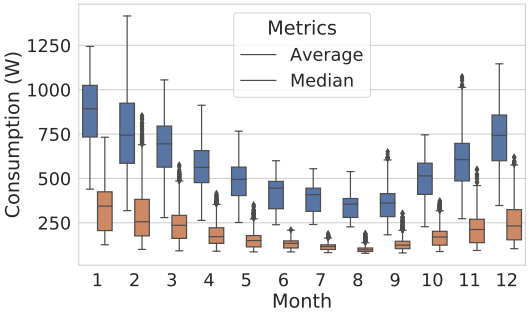


(f) 51 RES*+PRO*

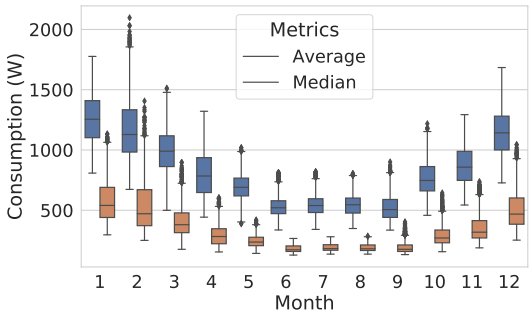
Figure B.15 – Standard deviation per 1/2h at regional levels



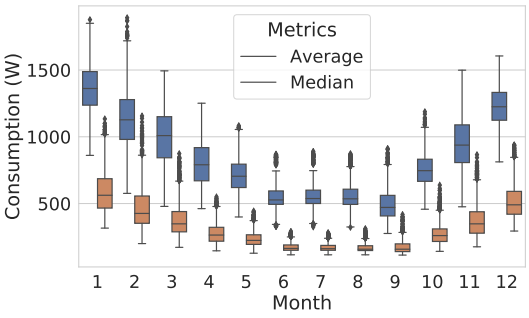
(a) National RES*+PRO*



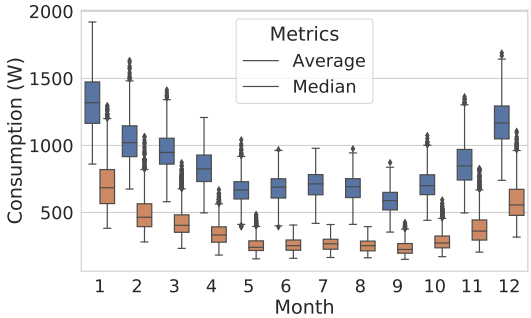
(b) 75 RES*+PRO*



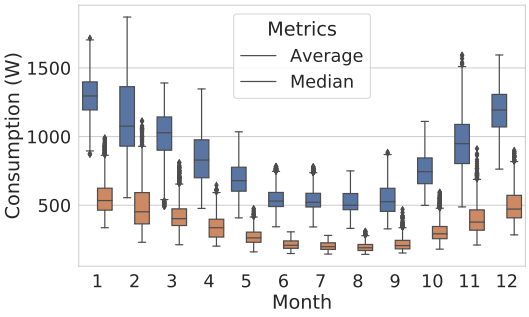
(c) 29 RES*+PRO*



(d) 74 RES*+PRO*

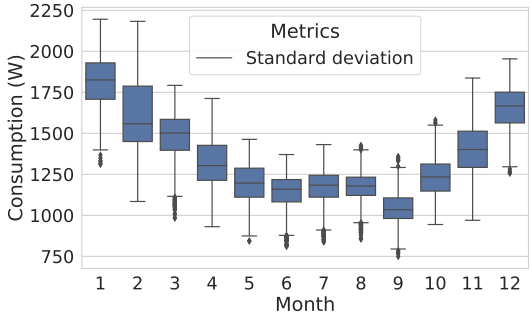


(e) 13 RES*+PRO*

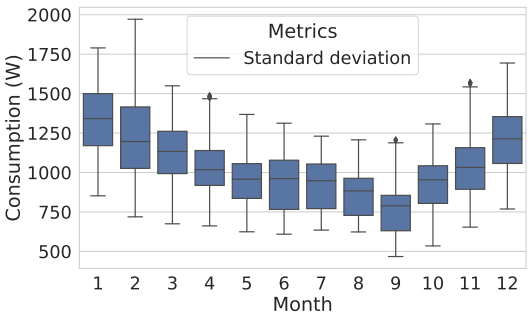


(f) 51 RES*+PRO*

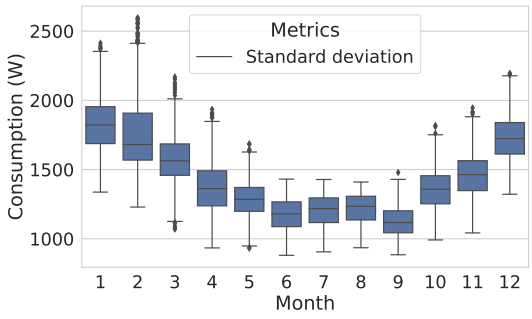
Figure B.16 – Average and median per month at regional levels



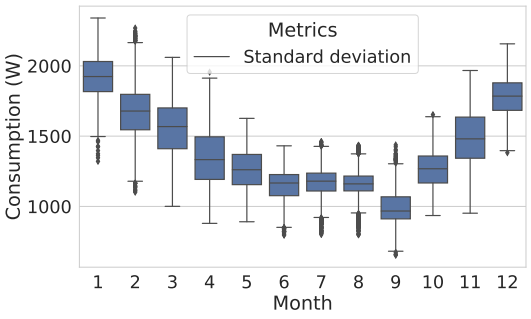
(a) National RES*+PRO*



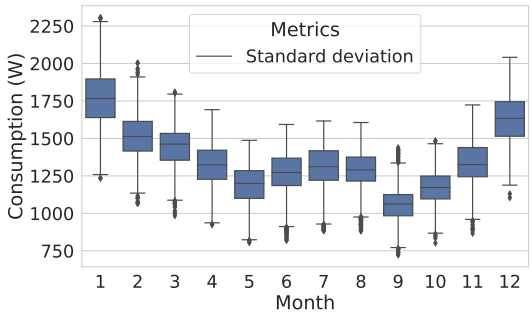
(b) 75 RES*+PRO*



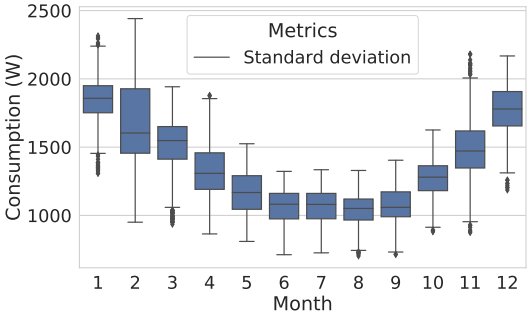
(c) 29 RES*+PRO*



(d) 74 RES*+PRO*

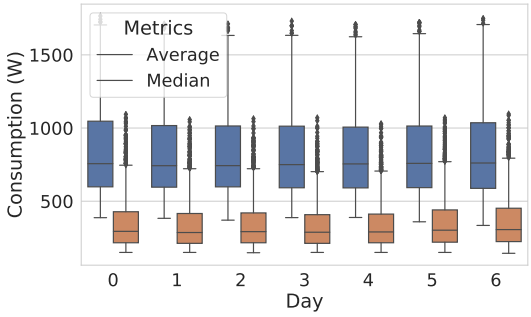


(e) 13 RES*+PRO*

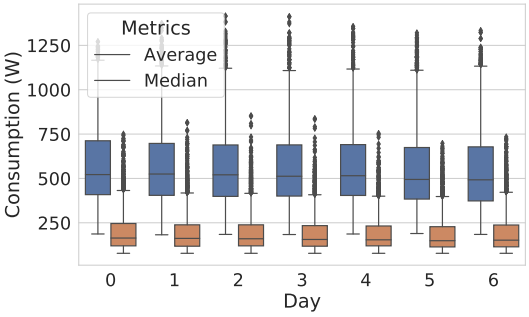


(f) 51 RES*+PRO*

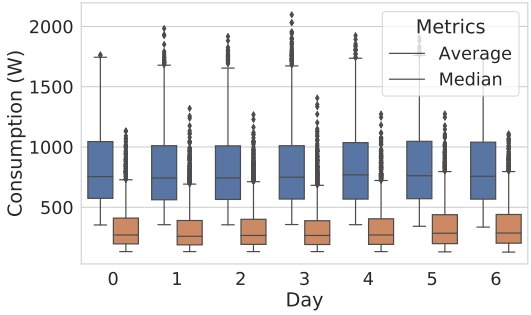
Figure B.17 – Standard deviation per month at regional levels



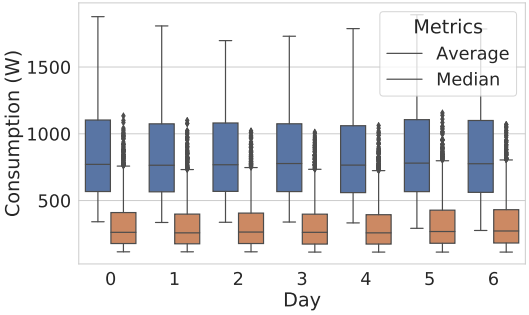
(a) National RES*+PRO*



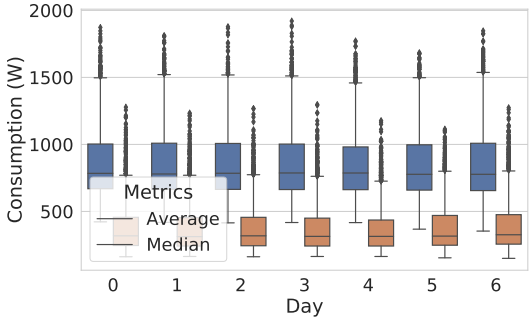
(b) 75 RES*+PRO*



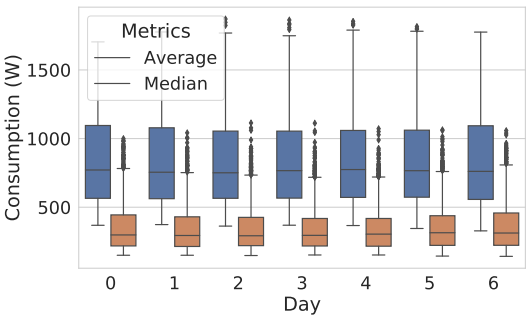
(c) 29 RES*+PRO*



(d) 74 RES*+PRO*

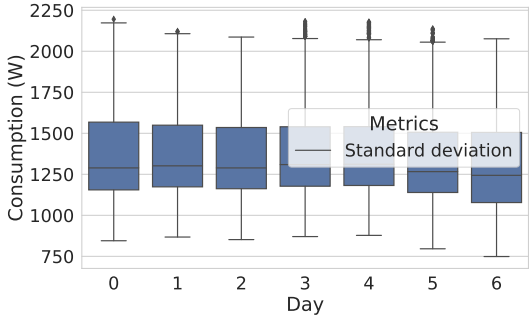


(e) 13 RES*+PRO*

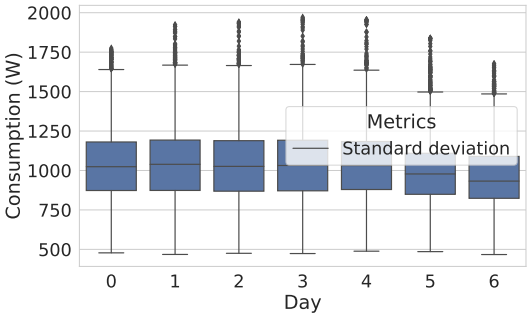


(f) 51 RES*+PRO*

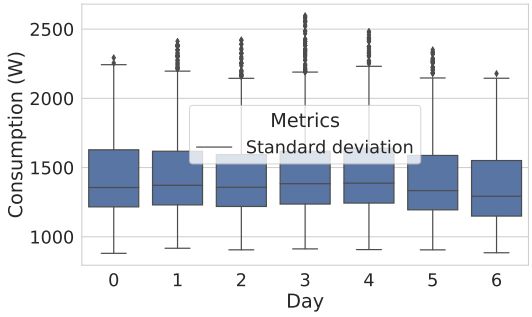
Figure B.18 – Average and median per week days at regional levels



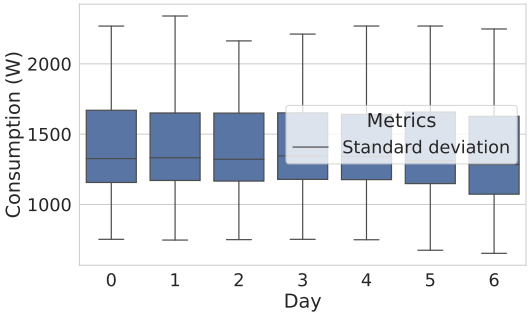
(a) National RES*+PRO*



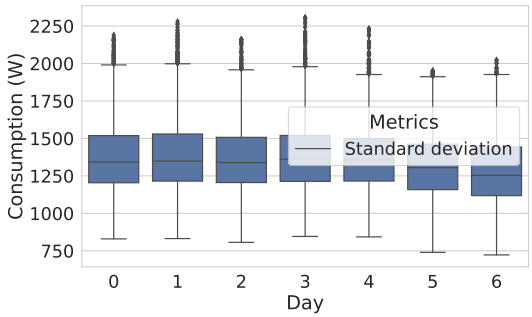
(b) 75 RES*+PRO*



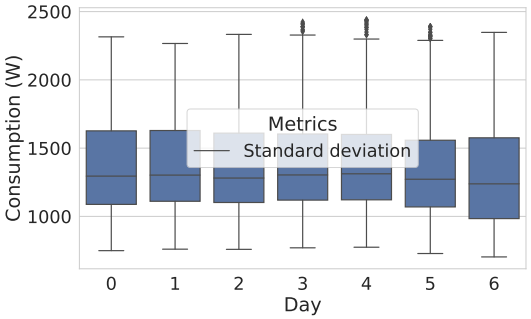
(c) 29 RES*+PRO*



(d) 74 RES*+PRO*

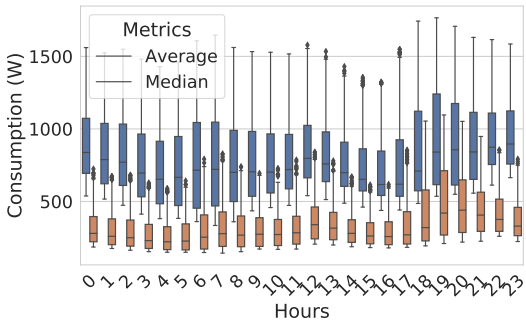


(e) 13 RES*+PRO*

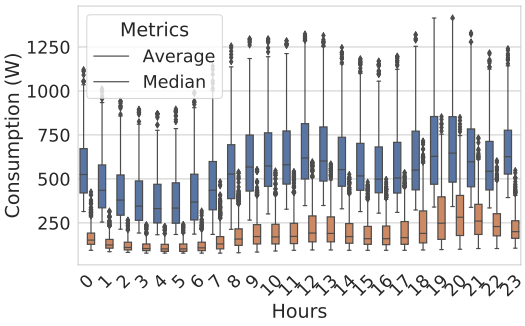


(f) 51 RES*+PRO*

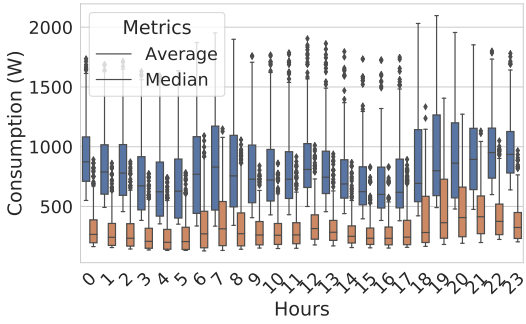
Figure B.19 – Standard deviation per week days at regional levels



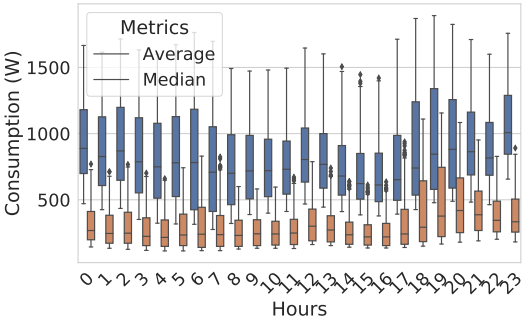
(a) National RES*+PRO*



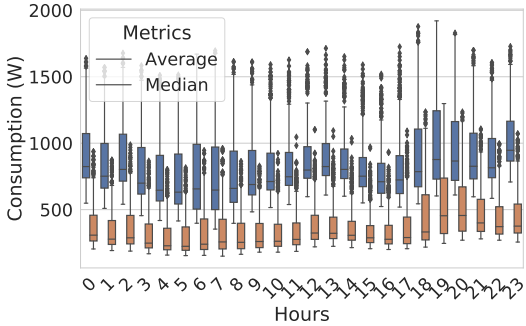
(b) 75 RES*+PRO*



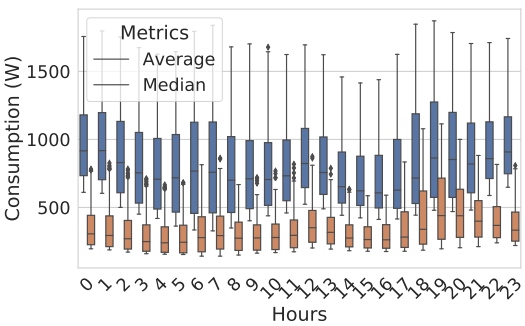
(c) 29 RES*+PRO*



(d) 74 RES*+PRO*

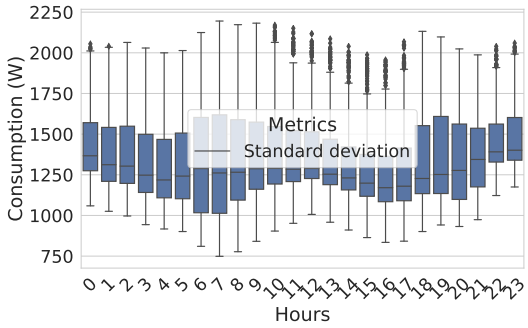


(e) 13 RES*+PRO*

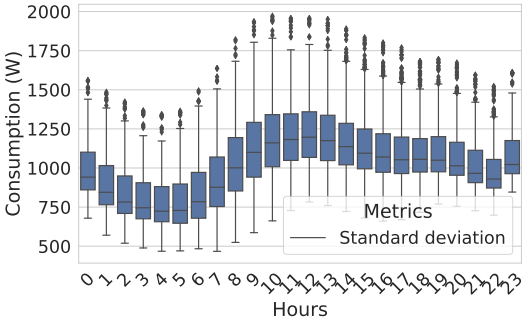


(f) 51 RES*+PRO*

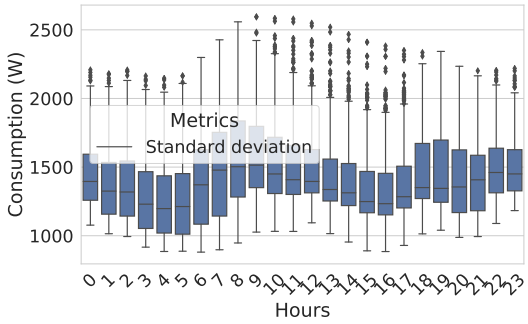
Figure B.20 – Average and median per hours at regional levels



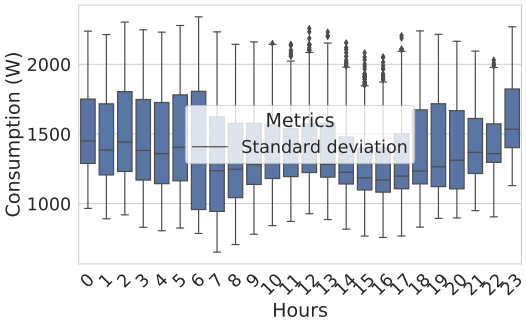
(a) National RES*+PRO*



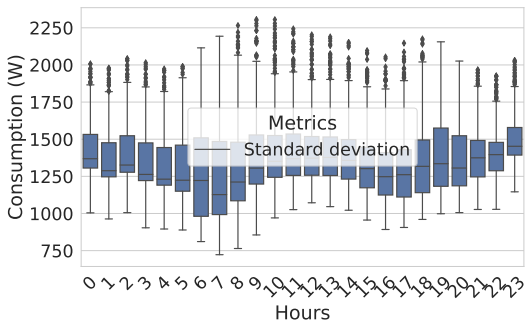
(b) 75 RES*+PRO*



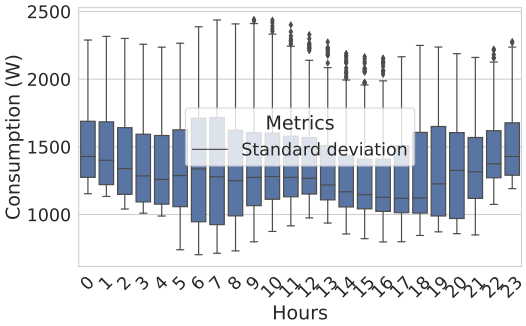
(c) 29 RES*+PRO*



(d) 74 RES*+PRO*



(e) 13 RES*+PRO*



(f) 51 RES*+PRO*

Figure B.21 – Standard deviation per hours at regional levels

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Analyse des Risques Liés à la Publication de Données Temporelles : Application aux Données de Consommation Électriques.

Mot clés : Vie privée, Données ouvertes, Consommation électriques

Résumé : Enedis est le principal gestionnaire de distribution d'électricité en France. Les distributeurs sont légalement obligé de mesurer et de publier la consommation électrique française. Les mesures contiennent de nombreuses informations personnelles et sensibles. De fait, la publication est anonymisée à l'aide d'agrégats par seuils. Ce travail étudie la vulnérabilité liée à la publication des mesures de consommation électrique. Notre première contribution est une étude statistique à grande échelle des mesures d'électricité française. En particulier, nous réalisons une étude d'unicité montrant que les séries non anonymisées sont très facilement identifiables. Notre deuxième contribution est une attaque par inférence d'appartenance qui permet de trouver toutes les séries formant un agrégat. Cette attaque est basée sur une variante du problème de la somme des sous-ensembles. Notre troisième contribution est une attaque par inférence d'appartenance modélisée comme un problème de classification de séries temporelles. Cette attaque nécessite peu de connaissances préalables et permet de trouver une cible spécifique dans un agrégat. Nous réalisons des expériences approfondies sur les attaques. Les résultats permettent de mieux choisir le seuil de publication. Enfin, nous proposons une méthode pour estimer la vulnérabilité des séries.

Title: Privacy Risk Analysis of Large-scale Temporal Data: Application to Electricity Consumption Data

Keywords: Privacy, Open data, Electricity consumption data

Abstract: The leading French electricity distribution manager, Enedis, legally must collect and publish electricity consumption time series. Series from households and companies are highly privacy sensitive. Therefore, the publication is anonymized using threshold aggregates. This work studies the vulnerability of open-sourcing electricity consumption time series. Our first contribution performs a large-scale statistical study of French electricity measurements. In particular, we perform a unique study showing un-anonymized series' high vulnerability against identification attacks. Our second contribution is a membership inference attack that finds every series forming an aggregate. This attack is based on a variant of the subset-sum problem. Our third contribution is a membership inference attack modelized as a time series classification problem. This attack requires little prior knowledge and can find a specific target in an aggregate. We perform in-depth experiments on the attacks. The results offer insight into the choice of relevant threshold. Finally, we propose a metric estimate the potential vulnerability of individual series.