5-2021

# Towards Secure Deep Neural Networks for Cyber-Physical Systems

Jiangnan Li
jli103@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Jiangnan Li entitled "Towards Secure Deep Neural Networks for Cyber-Physical Systems." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Engineering.

Jinyuan Sun, Major Professor

We have read this dissertation and recommend its acceptance:

Jinyuan Sun, Hairong Qi, Scott Ruoti, Lee D. Han

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

# Towards Secure Deep Neural Networks for Cyber-Physical Systems

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Jiangnan Li

May 2021

# Acknowledgments

I would like to express my sincere thanks to Dr. Jinyuan Sun, my advisor, for her patient guidance, help, and advice throughout my whole graduate study. Her support and insightful guidance are invaluable for me to complete this dissertation.

I wish to thank Dr. Hairong Qi, Dr. Scott Ruoti, and Dr. Lee D. Han for being on my committee members, reading my dissertation, and providing meticulous advice to improve the dissertation's quality. Meanwhile, I gratefully appreciate the financial support from CURENT and the Department of EECS of the University of Tennessee, Knoxville.

Last but not least, I would like to thank my colleagues and classmates: Yingyuan Yang, Xiangyu Niu, Eric Reinsmidt, Zhuo Yao, Yu Zhao, and Jin Young Lee for their help and support during my study.

# Abstract

In recent years, deep neural networks (DNNs) are increasingly investigated in the literature to be employed in cyber-physical systems (CPSs). DNNs own inherent advantages in complex pattern identifying and achieve state-of-the-art performances in many important CPS applications. However, DNN-based systems usually require large datasets for model training, which introduces new data management issues. Meanwhile, research in the computer vision domain demonstrated that the DNNs are highly vulnerable to adversarial examples. Therefore, the security risks of employing DNNs in CPSs applications are of concern.

In this dissertation, we study the security of employing DNNs in CPSs from both the data domain and learning domain. For the data domain, we study the data privacy issues of outsourcing the CPS data to cloud service providers (CSP). We design a space-efficient searchable symmetric encryption scheme that allows the user to query keywords over the encrypted CPS data that is stored in the cloud. After that, we study the security risks that adversarial machine learning (AML) can bring to the CPSs. Based on the attacker properties, we further separate AML in CPS into the customer domain and control domain. We analyze the DNN-based energy theft detection in advanced meter infrastructure as an example for customer domain attacks. The adversarial attacks to control domain CPS applications are more challenging and stringent. We then propose ConAML, a general AML framework that enables the attacker to generate adversarial examples under practical constraints. We evaluate the framework with three CPS applications in transportation systems, power grids, and water systems.

To mitigate the threat of adversarial attacks, more robust DNNs are required for critical CPSs. We summarize the defense requirements for CPS applications and evaluate several

typical defense mechanisms. For control domain adversarial attacks, we demonstrate that defensive methods like adversarial detection are not capable due to the practical attack requirements. We propose a random padding framework that can significantly increase the DNN robustness under adversarial attacks. The evaluation results show that our padding framework can reduce the effectiveness of adversarial examples in both customer domain and control domain applications.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

In recent years, new computational techniques have been studied to be employed in cyber-physical systems (CPSs). For example, the legacy power grid is on its way to becoming smart with diverse data-driven approaches. Smart grid, which introduces advancement in sensing, monitoring, control, and communication to legacy grids, is considered to be the next generation power grid that can provide high-quality service to the public [97]. The CPSs of critical infrastructures in modern society are usually complex and consist of enormous components. All these subsystems and components can become the sources of the large volume of miscellaneous data. Some data sources in the smart grid are shown in Fig. 1.1.

Enabled by the vast volume of data, machine learning (ML), especially deep neural networks (DNN), is increasingly studied in the research literature to be employed in different CPS applications. The DNN-based approaches have intrinsic advantages in learning the statistical patterns of the CPS data, which enables them to achieve state-of-the-art performances in many important applications, such as load forecasting in power systems [93, 51], energy theft detection of smart meters [116, 78, 49, 52, 119], incident detection [40, 62, 115, 87, 86, 120], cyberattack detection [83, 111, 44, 47, 10, 81, 105]. In addition, DNN systems are generally software-based and do not require extra equipment and device upgrades, which makes them compatible with the current CPS infrastructure.

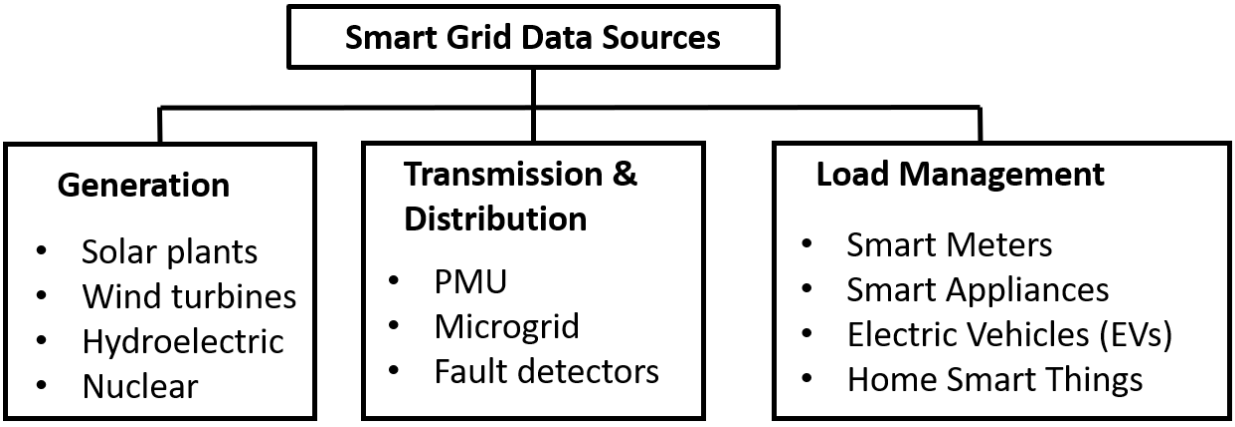


**Figure 1.1:** Data Sources in Smart Grid.

However, the deep learning techniques usually require large datasets to train DNN models, which brings new requirements for data management. Recent research suggests that the system operator outsource data management to third-part clouds for effectiveness and economic reasons [11]. Since the data may contain sensitive information of the system, the operator can encrypt the data before uploading them to the cloud. However, encryption will sacrifice the user's ability to query keywords over the data which is one of the common operations in CPS data management. Therefore, an effective searchable encryption scheme for CPS data is needed.

Meanwhile, as the DNN model parameters are automatically discovered during the learning process through back-propagation, DNN is considered as a black-box technique whose resulting computation is difficult to interpret, which brings great risk to employ the trained models in crucial applications, such as cyberattack detection in CPSs. Recent research on adversarial machine learning (AML) has demonstrated that well-trained DNN models are highly vulnerable to adversarial attacks. With the related AML algorithms, the attacker will be able to generate adversarial examples that deceive the DNN models to output the wrong results. Meanwhile, the same adversarial perturbations are transferable between different DNN models even the models are trained with different datasets and have different structures. Therefore, the potential vulnerabilities that adversarial attacks can bring to the CPSs need to be studied. However, the majority of AML research is conducted in the computer vision domain. Due to the inherent properties of the CPS applications, the widely-used threat model in previous AML research and state-of-the-art AML algorithms may become impractical, and new AML frameworks that are compatible with CPS applications are needed.

This dissertation mainly studies the power systems as the example of CPS applications, we also discuss other CPS applications, such as anomaly detection in water treatment systems, as different study cases to demonstrate the properties of our proposed approaches.

## 1.2 Challenges

### 1.2.1 Efficient Searchable Symmetric Encryption Scheme

Searchable symmetric encryption (SSE), which allows the user to query keywords over the cipher-text, draws attention in the cryptography research communities and several state-of-the-art SSE algorithms were proposed [94, 33, 17, 22]. However, the previous algorithms were designed for general plaintext encryption and may be inefficient for the CPSs data that was used as the DNN datasets. There are two properties that make the CPS data special for SSE. First, the data generated in the CPS applications, such as the smart grid, is frequently updated and owns a high generation rate. The newly generated data (i.e. new sensor measurements, new customer profiles) indicates that there are always new keywords generated, which makes the keyword dictionary-based SSE schemes impractical. In addition, different from the general plaintext, such as an email, the data generated by different CPS applications is usually well-regulated and follows specific structures. For example, the datasets used for DNN model training usually have constant feature numbers and the data of each feature owns the same data type. This property should be taken advantage of to design efficient SSE schemes for the CPS application.

### 1.2.2 Adversarial Attack and Defense in Cyber-Physical Systems

The adversarial attacks targeting DL applications in CPS can be quite different from attacks in pure cyberspace applications. For example, in the computer vision domain, the adversarial perturbations added to the legitimate input images should be as small as possible in order to avoid being noticed by human eyes, which is not applicable for CPS applications. In fact, the requirements of adversarial attacks in CPSs can also be different for different DL applications according to the attacker's resource, attack goals, and practical physical constraints. For example, DNN-based energy theft detection based on smart meter data has been studied in recent literature and achieves high detection accuracy [119, 78, 46, 70]. If an energy thief aims to steal energy by reporting false smart meter measurements to the utilities to make profit, she/he will need to focus on the total measurements instead of the divergent

(perturbation) between the real measurements and reported (adversarial) measurements. Currently, although there is research that studies adversarial attacks in CPSs [19, 65], it does not consider practical threat models and the specific attack requirements. Meanwhile, the defense mechanisms to mitigate the adversarial attacks in the CPSs have not been investigated.

## 1.3 Outline

The structure of this dissertation can be summarized as follow:

- In chapter 2, based on the properties of CPS data, we design an efficient searchable encryption scheme for CPS applications and achieves high space-efficiency. We implement a prototype based on the statistical data of advanced metering infrastructure in power systems to show the effectiveness of our approach.

- In chapter 3, we study the adversarial attacks in the customer domain of CPS. We investigate the vulnerability of the DNN-based energy theft detection and demonstrate that the well-perform DL models for energy theft detection are vulnerable to adversarial attacks. We design an adversarial measurement generation algorithm that enables the attacker to report extremely low power consumption measurements to the utilities while bypassing the DNN energy theft detection. The algorithm is evaluated with three kinds of neural networks based on a real-world smart meter dataset. The evaluation result demonstrates that our approach is able to significantly decrease the DNN models' detection accuracy, even for black-box attackers.

- In chapter 4, for the control system domain of CPS, we propose Constrained Adversarial Machine Learning (ConAML), a general AML framework for CPSs. We first summarize several practical constraints of AML in CPSs and formulate the mathematical model of ConAML by incorporating the physical constraints of the underlying system. We then design a series of AML algorithms that generate adversarial examples under the corresponding constraints. We evaluate the ConAML framework with three CPS applications, the incident detection in transportation

systems, the false data injection attack (FDIA) detection in power grid state estimation and the anomaly detection in water treatment systems. The evaluation results show that the adversarial examples generated by our algorithms can effectively bypass the DNN-powered attack detection systems.

- In chapter 5, we study the defense mechanisms against adversarial attacks in CPS. We analyze and evaluate several state-of-the-art adversarial defense mechanisms, such as adversarial training and adversarial detection, and demonstrate that they have intrinsic limitations for adversarial prevention in control domain adversarial attacks. We then design a robust DNN detection framework for FDIA by introducing random input padding in both the training and inference phases. We evaluate our framework with energy theft detection and FDIA detection. The results show that our framework greatly reduces the effectiveness of adversary examples in both customer domain and control domain applications.

# Chapter 2

# Space-Efficient SSE for CPSs

## 2.1 Introduction

Many cyber-physical systems (CPSs), such as the power grids, are critical infrastructure that contains miscellaneous data resources. Due to the divergence in structure, type and generation rate, how to integrate, store, and manage the data is still one of the active research fields in the research community. Recently, research on remote cloud-based storage and management of CPS data is becoming popular [11]. Arenas-Martinez *et al.* [7] presented and compared a series of cloud-based architectures to store and process smart meter reading data. Based on the specific characteristics of smart grid data, Rusitschka *et al.* [90] proposed a cloud computing model of ubiquitous data storage and access. In fact, outsourcing the data storage to the cloud can be an effective solution and has advantages in scalability, performance, and interoperability.

The cloud frameworks in [7] [90] work well when the CPS data owners own private cloud servers and or the cloud service is completely trusted. However, due to economic reasons, the utility company who owns the CPS data may choose to outsource the data storage to third-party cloud service providers (**CSP**). In this scenario, the cloud in the models becomes untrusted, and there is a privacy concern about the CPS data. Technically speaking, since the data are stored in the CSP's server, the provider can obtain full access to all the sensitive data easily. Moreover, it is reasonable to assume that the CSP may be interested in these data for many reasons. For example, the advanced metering infrastructure (AMI) data in

the smart grid can contain customers' personal information, such as hourly measured power consumption, home location, and payment record. By launching side-channel attacks, such as Non-intrusive load monitoring [41], it is possible for the CPS to learn the customer's habits and customs, which may bring profit to CPS in many ways.

One straightforward approach to prevent storage CPS from accessing sensitive data in the CPS is encrypting the data before uploading them to the cloud. While encryption provides confidentiality to the data, it also sacrifices the functionalities of processing the data. One of the most critical functions of processing data stored in the remote server is searching. For example, a utility company wants to query the billing statement of a specific customer. If the documents are encrypted with the concern of privacy, the CPS can no longer provide the search function to the utility company. In fact, this is a common problem that not only exists in the CPS field but also in all cloud storage applications that require privacy enhancement.

With the development of privacy-preserving technology, Searchable Symmetric Encryption (SSE) was proposed to address the above problem. SSE is technology that enables users to store documents in ciphertext form while keeping the functionality to search keywords in their documents. In recent years, a series of secure and efficient SSE schemes are proposed [94, 33, 17, 22, 50, 74, 68, 39, 14, 60]. However, most of them are only focusing on general circumstances in which user's documents are collections of random keyword combination, and can become inefficient or over-protect when directly applying them to CPS applications, such as smart grid.

There are two characteristics that make CPS data special. Firstly, CPS data are believed to be frequently updated and have a high generation rate. This also implies there are always new data/keywords generated, which will lead to an increasing keyword dictionary. Furthermore, a large portion of CPS data are well regulated and have specific structures. In practice, CPS data may contain multiple attributes that will be searched as keywords. For example, although two utility companies may have different implementations of storing customer billing statements, it is reasonable to assume both implementations should have keywords such as user identity, electricity price, smart meter reading in each record. This assumption is also practical for the datasets used for ML applications in the CPS since the datasets are generally well-regulated and each record in the dataset contains the same number

of features. The two characteristics make the state-of-the-art SSE schemes inappropriate to be applied to CPS applications. The typical SSE schemes together with their disadvantages with the above two characteristics will be discussed in Section 2.2.

The contributions of this chapter are summarised as follow:

- We review and analyze the typical state-of-the-art SSE schemes and show why they are inappropriate for CPS data.
- According to the characteristics of CPS data, we design a simple, practical SSE scheme that provides higher space efficiency with tolerant information leakage in real applications.
- We implement a prototype based on the statistic data of advanced metering infrastructure (AMI) provided by U.S. Energy Information Administration (EIA) to show the effectiveness of our scheme. We claim the scheme can also be applied to other types of data in CPSs, such as metering data, customer billing statement, and PMU/PDC data in smart grids.

### 2.1.1 Outlines

The rest of chapter is organized as follow. Section 2.2 gives the introduction of related work on SSE. Section 2.3 introduces the threat model and notations we use. The detailed description of our scheme will be presented in Section 2.4. After that, Section 2.5 introduces the implementation of the prototype and smart grid data we use.

## 2.2 Related Work

Currently, the fundamental SSE constructions can be classified into three categories, namely construction without indexes, construction with direct index, and construction with inverted indexes.

The first practical SSE scheme without index construction was proposed by Song *et al.* in 2000 [94]. They considered a document as a list of words with the same length and used

a specially designed stream cipher to encrypt the document. However, this scheme requires the server to traverse each document word by word, which leads to a search complexity linear to the document size. Furthermore, the SSE scheme without index usually requires specially developed encryption algorithms, making it unscalable to current CPS communication and control systems. After that, several high impact index-based SSE schemes were proposed. Secure Indexes by Goh [33] built Bloom Filters as the direct index. By adjusting the parameters of the Bloom Filter, secure indexes can achieve efficient search complexity. However, one inherent problem of Bloom Filter is that it will bring a false-positive rate, and this can be unacceptable for critical infrastructure CPSs, such as the smart grid. Another direct index-based scheme was presented by Chang *et al.* [17], and they built a large index table for all documents to enable the efficient search. However, their scheme assumes that there is a dictionary mapping all keywords to associate identifiers. As discussed in Section 2.1, the number of keywords in CPS data can be large and keep increasing. Therefore, the scheme in [17] is also not appropriate for CPS applications. One of the most famous inverted index based SSE schemes were proposed by Curtmola *et al.* [22]. They presented an indexing scheme that achieves the highest time-efficient search function by using a uniquely designed linked list data structure. However, an inverted index construction scheme has an inherent problem, namely directly updating is difficult. Although a well-designed file management system can mitigate the problem, inverted indexes are still not efficient for the CPS which has new data generated all the time. The latest research work on SSE including dynamic searchable encryption [50, 74, 68, 39], forward secure searchable encryption [14], and fuzzy keyword searchable encryption [60].

## 2.3 Background

Besides the theoretical of the SSE scheme, research on applying SSE to solve real-world problems is also active. In general, the SSE scheme can be used to all systems which include storage outsourcing. Tong *et al.* [102] employed a modified SSE scheme of Secure Index [33] and designed a secure data sharing mechanism for situational awareness in the power grid. Their approach is also used to protect the privacy of e-health data [103]. The problem of

health data privacy is drawing more and more attention. Li *et al.* [57] leveraged a secure K-nearest neighbor (KNN) and attribute-based encryption to build a dynamic SSE scheme for e-health data and achieves both forward and backward security. Other applications of SSE can be found in [13].

### 2.3.1 Threat Model and Assumptions

Our scheme interacts between two parties. As shown in Fig. 2.1, we refer the party who owns the data and wants to outsource the storage of data as the client, and the party who provides the storage service as the server. The client uploads the encrypted data, and associated search index to the server and sends query afterward. For simplicity, our scheme only considers the case that there is one keyword contained in the query. We note that a query that contains boolean operation on multiple keywords can be regarded as the operation on query results of multiple single keywords, which will be briefly discussed at the end of this section. After receiving the query contains keyword information from the client, the server should run the SSE algorithms and return a list of the identifier of documents that contain the keyword.

Same as most state-of-the-art SSE schemes, in our threat model we also assume that the third party server is a curious but honest attacker, which means the cloud server should provide normal cloud service but will try to learn the content of data. More specifically, the server is not allowed to delete or modify the client's data or share the data with other parties. However, different from SSE schemes which assume that the client's data is a collection of random keyword combinations, our scheme makes practical assumptions on the client's data based on the characteristics of smart grid data.

First, we assume the CPS data to be stored in the cloud should be a collection of records that have the same data structure and contain the same number of keywords. For example, the customer billing statement record in advanced metering infrastructure should have attributes like customer identifier, date, house location, smart meter readings, and additional notes.

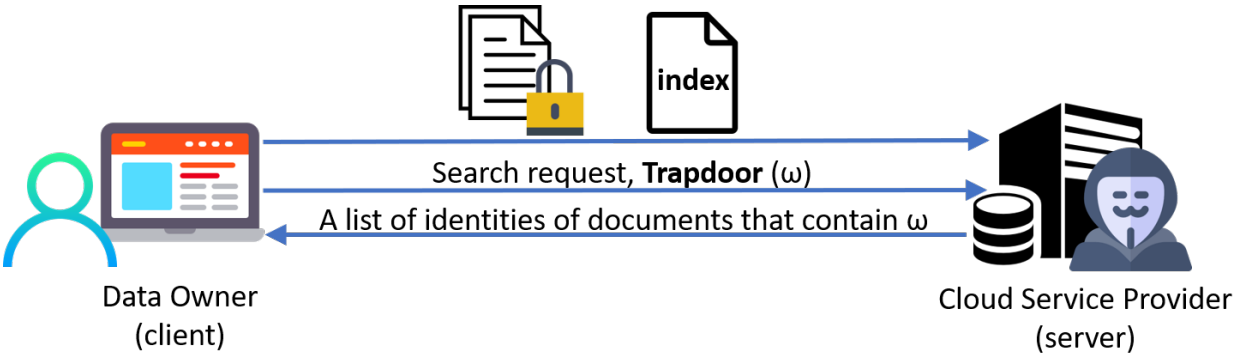


**Figure 2.1:** General SSE Working Model.

Another example can be the PMU data. One of the most widely used standards of PMU data is IEEE C37.118.2 [1]. The standard gives the format of the application layer data structure of PMU data that contains several attributes, such as data stream ID number, time stamp, and measurement data. Moreover, as presented in [102], the data owner can also design a hierarchical data structure to store the CPS data. Therefore, we believe this assumption is practical in real CPS applications and is easy to meet. In the rest of the chapter, we will use record and document interchangeably for simplicity.

Second, we assume the total number of records to be stored in the cloud is much larger than the number of keywords in each record. In practice, a record may contain up to tens of attributes, but it is normal for a utility company to maintain millions of customer data records or billions of measurement data records.

One widely used security requirement of searchable symmetric encryption is IND2-CKA secure proposed in [33]. In brief, IND2-CKA secure requires that the server cannot learn anything more of the plaintext message except for the search result. According to IND2-CKA, the number of keywords in each document should also be kept secret. However, our scheme allows the number of keywords in each record to be known to the attacker (server). There are mainly two practical reasons for this privacy sacrifice. First, SSE schemes follow IND2-CKA secure consider the case that the client stores general documents which contain keywords with random numbers. As we discussed above, since all CPS data records are assumed to contain the same number of keywords, it is not reasonable to assume that the attacker cannot learn the number in the real working scenario, like by social engineering or just randomly guessing. Second, we will show that our scheme can achieve high space efficiency by losing the restriction on this privacy leakage issue. This compromise should be acceptable for a utility company or government department in practice.

Finally, the scheme is only used for constructing a secure search index. How to protect the privacy of the plaintext data will not be discussed in this chapter. The client can use popular security schemes like CBC or CTR block cipher to protect the confidentiality and schemes like HMAC to protect the integrity of data.

### 2.3.2 Notations

We use $\Delta = \{R_1, R_2 ... R_{2^d}\}$ to indicate a collection of records. $N$ is defined to be the total number of possible keywords in a record while $n$ is the number of desired keywords in a record that the client want to search, where $n \leq N$. We use $w_{i,j}$ to denote the $jth$ desired keyword in record $R_j$ where $1 \leq i \leq 2^d$ and $1 \leq j \leq n$. $\{0,1\}^n$ is used to present the set of all $n$ bits numbers. $K \xleftarrow{R} \{0,1\}^n$ means an element $K$ being sampled uniformly from set $\{0,1\}^n$. In addition, we use $f : \{0,1\}^k \times \{0,1\}^r \rightarrow \{0,1\}^r$ to define a pseudo-random function that maps a $r$ bits number to another $r$ number with a $k$ bits key. A record's identifier is defined as $id(R)$. Finally, we use $h : \{0,1\}^* \rightarrow \{0,1\}^r$ to denote a hash function that maps random length bitwise string to an $r$ bits number.

## 2.4 Practical SSE Scheme Design For Cyber-Physical Systems

### 2.4.1 Construction

Secure Indexes presented in [33] gives a framework of trapdoor based searchable symmetric encryption scheme which was widely used by the follow-up research. In general, an SSE should consist of four polynomial time algorithms:

- **Keygen($s$)** is run by the client to generate a master private key $MK$ where $s$ is a security parameter.
- **Trapdoor($MK$, $w$)** is run by the client by taking the master key $MK$ and a keyword $w$ as the input, and outputs the trapdoor $T_w$ of word $w$.
- **BuildIndex($R$, $MK$)** is run by the client by taking the master key $MK$ and a record $R$ as the input, and outputs the index $I_R$ for record $R$.
- **SearchIndex($T_w$, $I_R$)** is run by the server by taking a trapdoor $T_w$ and a document's index $I_R$ as the input, and outputs 1 if $w \in R$ or 0 otherwise.

In general SSE scheme, the encrypted documents together with the associated indexes will be kept by the server. When searching, the client generates the trapdoor $T_w$ for word $w$ and send $T_w$ to the server. For each record $R$, the server runs the **SearchIndex($T_w$, $I_R$)** function and determine whether $R$ contains $w$. The server will finally return to the client a list of the identifiers of records which contain $w$. The framework requires that only the client who holds the private master key $MK$ can generate the trapdoor $T_w$ for each word $w$, such that the server cannot learn related information from the index. Our scheme also follows this framework and is built in the direct index structure.

Our scheme uses a codeword array as the index for a record. For the keyword $w$, the **Trapdoor** function computes the hash value $h(w)$ of $w$, where $h : \{0,1\}^* \rightarrow \{0,1\}^r$, and outputs the trapdoor $T_w = f_{MK}(h(w))$, where $f : \{0,1\}^k \times \{0,1\}^r \rightarrow \{0,1\}^r$ is a pseudo-random function. To build the index $I_{R_i}$ of record $R_i$, the **BuildIndex** function calls **Trapdoor** and computes the codeword $X_{i,j} = f'_{T_{w_{i,j}}}(h(id(R_i)))$, where $f' : \{0,1\}^r \times \{0,1\}^r \rightarrow \{0,1\}^r$ is another pseudo-random function. Each codeword $X_{i,j}$ should be randomly written into an n element array $I_{R_i}$, and this step can be done with a pseudo-random generator in implementation. The detailed design of our scheme can be found in Fig. 3.

As discussed early in this section, we assume the CPS data $\Delta$ to be a collection of records with each record $R_i$ contains $N$ keywords. Considering the scenario that not all types of keywords in the data are necessary for searching, and the data owner only wants to query records by several specific types of keywords. For example, based on the standard IEEE C37.118.2, the utility company may want to search PMU data records by ID number or timestamp, and there may be no need to search by synchronization word. Therefore, our scheme firstly allows the data owner to select the $n\,(n \leq N)$ types of keywords she wants to query based on specific applications.

After determining $n$ types of keywords, the client should run function **Keygen** to obtain a $k$ bits master key $MK$ and keep it secret. Subsequently, for each record $R_i$ in $\Delta$, the client runs the **BuildIndex** function to obtain the index $I_{R_i}$ of record $R$. Finally, the index $I_{R_i}$ should be attached to the encrypted record $R_i$ and uploaded to the server.

To search for a keyword $w$, the client needs to compute the trapdoor $T_w = f_{MK}(h(w))$ and sends $T_w$ to the server. After receiving the $T_w$, the server computes $X_{i,w} = f'_{T_w}(h(id(R_i)))$

- **Keygen($k$)**: Uniformly sample the master key $MK \xleftarrow{R} \{0,1\}^k$
- **Trapdoor($MK, w$)**: Given $MK$ and keyword $w$, generate $T_w = f_{MK}(h(w))$ of $w$
- **BuildIndex($R_i$, $MK$)** : The input is a record $R_i$ and the master key $MK$

  **The client:**

  create an $n$ elements array $I_{R_i}$ and initialize all elements to zero
  create an set $U : \{x \in \mathbb{Z} | 0 \leq x \leq n-1\}$
  **for** *each desired keyword* $w_{i,j}$ *in* $R_i$ **do**
      compute $T_{w_{i,j}} = \textbf{Trapdoor}(MK, w_{i,j})$
      compute $X_{w_{i,j}} = f'_{T_{w_{i,j}}}(h(id(R_i)))$
      pick $\lambda \xleftarrow{R} U$, update $U = U - \{\lambda\}$
      set $I_{R_i}[\lambda] = X_{i,j}$
  **end**
  return $(R_i, I_i)$

- **Search($T_w, I_R$)** : Given $T_w$ and $I_R$ return search result.

  **The server:**

  compute $X_w = f'_{T_w}(h(id(R)))$
  **if** $X_w$ *is in list* $I_R$ **then**
      return 1
  **else**
      return 0
  **end**

**Figure 2.2:** Four Polynomial time functions

for each ciphertext record $R_i (1 \leq i \leq 2^d)$, and checks whether $X_{i,w}$ is contained in $I_{R_i}$. If so, the server returns $id(R_i)$ to the client.

### 2.4.2 Analysis and Comparison

As stated at the beginning of Section 2.3, our scheme aims to protect the data privacy such that the adversary cannot learn any other information about the plaintext record from the index except for the search result and the number of desired keywords in the record. We analyze the security of our scheme from two aspects. First, considering a simple scenario that only one index of a record is given, a polynomial-time attacker can not learn the original keywords from the index. This is correct because if the attacker can learn the keyword from the codeword, she will be able to break the pseudo-random function, which is contradictory to the assumptions. Second, we consider the unlinkability of our scheme, which means the attacker is not able to learn whether two records have the same keyword $w$ from their indexes without the trapdoor $T_w$. This is achieved by introducing the identifier of records to build the codeword. The same keyword in two records will have different codewords in two indexes. We refer the reader to [33] for the mathematical proof.

In general, the main methodology of our scheme is increasing the space efficiency of the SSE scheme with the permission of a few information leakages based on the characteristics of CPS data. Our scheme was built based on the direct index structure, so it is dynamic and easy-updating. We use the codeword array as the index of each record, which leads to a small index size compared with schemes that involve the keyword dictionary. Since the codewords are randomly inserted, the searching complexity of our scheme becomes $\mathcal{O}(2^d \cdot n)$. However, since the number of desired keywords $n$ is believed to be a very small constant in practical application, the search algorithm will still be efficient. Our scheme is similar to the PPSED scheme described in [17]. Both schemes used direct index structure and an array as the index. However, since there are always new keywords (e.g. timestamps) generated in CPS data, the index size of the PPSED scheme will become extremely large. Table 2.1 gives a detailed comparison between our design and the widespread SSE schemes.

### 2.4.3 Extension

It is obvious to see that updating the new record and associated index to the server is straightforward. The client just needs to run the **BuildIndex** function of the new record to obtain the index, and appends the (encrypted record, index) pair to the records and indexes stored in the server.

The plain scheme considers the scenario that only one keyword is searched in a round. In practice, the client may want to search for records that meet specific keyword requirements. One simple example can be a government department that wants to query a dataset of utility companies. A meaningful query can be (State: TN and Establish_Year: 1998 or 1999). The boolean operations on multiple keywords can be easily applied to the codewords matching process when the server runs **Search** function.

## 2.5 Implementation

### 2.5.1 Example Smart Grid Data

The public dataset we used to test the effectiveness of our scheme is the statistical AMI data provided by the U.S. Energy Information Administration **(EIA)** [4]. The AMI data are derived from EIA-861M form, which stands for "Monthly Electric Power Industry Report." The report collects sales of electricity and revenue each month from a statistically chosen sample of electric utilities in the United States. EIA started to collect monthly green pricing, net metering, and advanced metering data since 2011. We choose the CSV file of advanced metering data of the year 2016 as the dataset for our implementation.

As shown in Table 2.2, the CSV contains 31 columns, including year, month, utility name, state, residential AMI, and so on. In our experiment, we select all features except Year from Utility Characteristics and all features from AMI related categories as our desired types of keywords. The CSV file contains 4819 records in total, and the data types include string and unsigned integer.

**Table 2.1:** Perfomance Comparison of Various SSE Schemes

| **scheme** | **Encryption** | **index** | **FP** | **update** | **Complexity** | **Size** |
|---|---|---|---|---|---|---|
| final scheme [94] | special | no | no | easy | $\mathcal{O}(2^d \cdot N)$ | none |
| Z-IDX [33] | general | direct | yes | easy | $\mathcal{O}(2^d)$ | $\mathcal{O}(2^d \cdot n)$ |
| PPSED [17] | general | direct | no | easy | $\mathcal{O}(2^d)$ | $\mathcal{O}((2^d)^2)$ |
| SSE-1 [22] | general | inverted | no | hard | $\mathcal{O}(1)$ | $\mathcal{O}(2^d \cdot n)$ |
| our scheme | general | direct | no | easy | $\mathcal{O}(2^d \cdot n)$ | $\mathcal{O}(2^d \cdot n)$ |

$2^d$is the number of records, $N$ is the keywords number in a record.
$n$ is the number of desired keywords in a record, where $2^d \gg n$.

### 2.5.2 Prototype

For simplicity, we use Python 2.7 as the programming language for the prototype. We claim that a low-level language like C can be used in practice with the consideration of speed. We build the encryption scheme with the help of Pycrypto [2], which is a widely used cryptography library for python. Generally, the ciphers and hash functions in Pycrypto are written in C and provide Python API. We use Advanced Encryption Standard **(AES)** block cipher with 128 bits key as the pseudo-random function and the MD5 as the hash function described in our scheme. We note that MD5 has been severely compromised and is no longer secure for integrity protection. However, the MD5 in our prototype is only used to generate a unique identifier for record and keyword, similar to the building dictionary process in other schemes.

Since the original ciphertext usually contains unprintable characters that will destroy CSV format, we store the ciphertext as the hexadecimal string to maintain a clear CSV format for demonstration. However, the hexadecimal string will lead to a larger index size. Therefore, we suggest that an efficient file-index management system is needed for real-world applications.

The source code is tested on a Mac OS X machine with a 1.6 GHz Intel Core i5 processor and 8GB memory. Our experiment result shows that 4819 indexes can be searched in around 0.15 seconds. The source code of the prototype is available on Github [3].

**Table 2.2:** Features of AMI Dataset from EIA

| **category** | **feature** |
|---|---|
| Utility Characteristics | Year, Month, Utility Number, Utility Name, State, Data Status |
| Number AMR - Automated Metering Reading | Residential, Commercial, Industrial, Transportation, Total |
| Number AMI - Advanced Metering Infrastructure | |
| Non AMR/AMI Meters | |
| Total Number of Meters | |
| Energy Saved - AMI (MWh) | |

# Chapter 3

# Customer Domain Adversarial Attacks: Energy Theft Detection

In this chapter, we study the adversarial attack in DNN-based energy theft detection as an example of the customer domain of cyber-physical systems.

## 3.1 Introduction

Energy theft causes high financial losses to electric utility companies around the world [31]. In recent years, two-way data communications between the customers and utilities are enabled by the development of the advanced metering infrastructure (AMI). Smart meters that provide fine-grained power consumption data of customers are expected to mitigate energy theft. However, the smart meters are shown to be vulnerable to physical penetration [9] and there are even video tutorials online on smart meter hacking [95]. To date, the energy theft problem is still serious and the corresponding detection approaches are needed.

Currently, the energy theft detection approaches proposed in the literature can be categorized into sensor-based and user profile-based detection. The sensor-based methods requires extra equipment to be deployed in AMI while the user profile-based detection exploits the abnormal variations in customer's power usage patterns. In recent years, machine learning (ML) techniques, especially deep learning (DL), are studied in the literature to detect energy theft [118, 46, 116, 31, 78, 49, 85, 119, 42, 52]. The ML-based approaches

take advantage of the massive fine-grained power consumption data of smart meters and can achieve state-of-the-art performance. Meanwhile, they are usually pure software systems and are compatible with the legacy power system infrastructures.

However, recent studies in the computer vision (CV) domain have shown that the well-trained DL models are highly vulnerable to adversarial examples [96, 35, 89, 77**?** ]. By adding a well-crafted perturbation to the legitimate input, adversarial attackers can deceive the DL models to output wrong prediction results. The same perturbation is transferable between different DL models that own different structures and are trained with different datasets. Adversarial attacks are also demonstrated to be effective in power system applications [19, 20, 59]. As DL becomes a popular technique in energy theft detection, the potential risks of adversarial attacks need to be investigated, which is the focus of this paper.

Although sophisticated adversarial machine learning (AML) algorithms have been proposed in the CV domain, they can not be applied for energy theft directly due to different requirements for the examples. For instance, the adversarial perturbations in the CV domain are required to be small so that the adversarial image will be hardly perceptible to human eyes. Since such constraints do not apply to an energy thief, the performances of the AML algorithms in energy theft need to be evaluated. To increase the stolen profit, the energy thief should focus on the size of the adversarial example (power consumption measurements reported to the utilities) instead of the perturbation. The adversarial attack that maximizes the attacker's profit should be formulated to evaluate the reliability of the DL detection models.

In this chapter, we investigate the vulnerabilities of DL-based energy theft detection through AML, including single-step attacks and iterative (multiple-step) attacks. We design *SearchFromFree*, a framework to increase the attacker's profit. The main contributions of this chapter can be summarized as follows:

- We study the vulnerabilities of DL-based energy theft detection and summarize the properties of adversarial attacks in energy theft detection by proposing a general threat model.

- We propose a random adversarial measurement initialization approach to maximize the attacker's stolen profit. It is compatible with different state-of-the-art AML algorithms and can generate valid adversarial examples with low energy costs. Meanwhile, we design an iterative adversarial measurement generation algorithm that employs a step-size search scheme to increase the performance of black-box attacks.

- The evaluations are implemented with three types of neural networks that are trained with a real-world smart meter dataset. The result shows that our framework can generate small adversarial measurements that can successfully bypass the detection.

## 3.2 Related Work

The support vector machine (SVM) was employed by Nagi *et al.* to detect abnormal power usage behaviors based on the historian consumption data in 2009 [79]. Depuru *et al.* extended their approach and included more features, such as the type of consumer and geographic location [24]. Jokar *et al.* generated a synthetic attack dataset and trained a multi-class SVM classifier for each customer to detect malicious power consumption [49]. SVM is also combined with other techniques, such as a fuzzy inference system [80] or decision tree [48] to detect energy theft. In 2017, Zheng *et al.* employed deep convolutional neural networks (CNN) to detect energy theft based on a real-world dataset and achieved a high detection rate [119]. [78] trained a deep recurrent neural network (RNN) and randomly searched for model parameters. In 2020, Ismail *et al.* studied energy theft in the distributed generation domain and proposed a hybrid neural network detection model [46]. Other DL-based energy theft detection approaches can be found in [42, 52].

In 2013, Szegedy *et al.* proposed the adversarial attacks to deep neural networks in the CV domain [96]. After that, various AML algorithms were proposed, such as the Fast Gradient Sign Method (FGSM) by Goodfellow *et al.* [35], Fast Gradient Value (FGV) by Rozsa *et al.* [89], and DeepFool by Moosavi-Dezfooli *et al.* [77]. Recently, adversarial attacks on power system ML applications are also investigated. Chen *et al.* showed that both the classification and regression applications in the power system are vulnerable to adversarial attacks [19]. They then launched adversarial attacks to study the vulnerabilities of load

forecasting [20]. In 2019, Liu *et al.* showed that the DL-based AC state estimation can be compromised by adversarial attacks [65]. [59] studied the DL-based false data injection attack detection in DC state estimation and demonstrated that the attacker can compromise both the DL detection and residual-based detection with physical constraints. Marulli *et al.* studied the data poisoning attacks to ML models in energy theft detection using the generative adversarial network (GAN) [71] but they did not consider the evasion attacks and the attacker's profit.

## 3.3 Formation and Design

### 3.3.1 Adversarial Energy Theft Formation

To launch energy theft, the attacker is assumed to be able to compromise his/her smart meter and freely modify the power consumption measurements that are reported to the utilities. In general, the DL-based energy theft detection in AMI can be considered as a binary classification problem. Given the power consumption measurements $M$, the utility company utilizes a DL classifier $f_\theta : M \rightarrow Y$ trained by a dataset $\{M, Y\}$ to map the measurements $M$ to their labels $Y$ (Normal or Theft). The adversarial attack in energy theft detection should be a false-negative attack that deceives the DL classifier $f_\theta$ to categorize the adversarial measurement vectors $A$ as normal. Meanwhile, $A$ needs to be small so that the energy thief can obtain a high profit. Without loss of generality, the adversarial attack in energy theft can be represented as an optimization problem:

$$\min \quad \|a\|_1 \tag{3.1a}$$

$$s.t. \quad f_\theta(a) \rightarrow Normal \tag{3.1b}$$

$$a_i \geq 0 \tag{3.1c}$$

where $a$ represents an adversarial measurement vector, $\|a\|_1 = \sum \|a_i\|$ is the $L_1$-Norm of $a$. The constraint (3.1c) requires that all the power consumption measurement $a_i$ in $a$ must be non-negative to be feasible.

### 3.3.2 Threat Model

We propose a practical threat model for the adversarial attacks in energy theft detection, as described below:

- The attacker can freely modify the meter's power consumption measurements reported to the utilities. In practice, this can be implemented through physical penetration to the smart meter.
- If a DL model was trained by the utilities, it will usually be deployed on a separate server that owns isolated access networks. We consider a black-box adversarial attack that the energy thief cannot access to the utilities' DL model $f_\theta$ and training dataset $\{M, Y\}$.
- The attacker can obtain an alternative dataset $\{M', Y'\}$, such as a historian or public dataset, to train his/her model $f'_{\theta'}$ to generate adversarial measurements. The principle behinds the black-box attack is the transferability of adversarial examples.
- The attacker needs to generate non-negative adversarial measurements to be practical, as shown in 3.1c.

In section 3.4, we will also evaluate the performance of white-box attacks that allow the attacker to fully access the DL model $f_\theta$, such as insider attackers. Such evaluations can study the reliability of the detection system under the worst-case scenario and the upper bound performance of the adversarial attacks.

### 3.3.3 State-of-the-art Approaches

Since the constraint (3.1b) defined by the neural network is highly-nonlinear, formation (3.1) is difficult to solve directly by existing optimization approaches. Generally, the existing AML

algorithms maximize the adversarial attack performance by increasing the prediction loss of the DL model through gradient-based optimization. We release the related constraints, such as the box-constraint, in the CV applications to fit the energy theft attack requirements.

In general, the AML algorithms can be categorized into single-step attacks and iterative (multiple-step) attacks. The single-step attacks usually have better transferability but are relatively easy to defend, while the iterative attacks are more powerful but are less transferable [54]. In this paper, we study three state-of-the-art AML algorithms, FGSM [35], FGV [89] and DeepFool [77], as shown below:

**Fast Gradient Sign Method (FGSM)**

The FGSM method proposed in [35] is a single-step attack method. Given $a$, FGSM updates the vector according to equation (3.2), where $\epsilon$ is a constant, $L$ is a loss function, and $Y_a$ is the label (theft) of $a$. With the *sign* function, the perturbation vector size is controlled by $\epsilon$.

$$a = a + \epsilon \cdot sign(\nabla_a L(f_\theta(a), Y_a)) \tag{3.2}$$

**Fast Gradient Value (FGV)**

The FGV method proposed in [89] is also a single-step algorithm and is similar to FGSM. However, FGV employs the original gradient values instead of the sign value, as shown in equation (3.3).

$$a = a + \epsilon \cdot \nabla_a L(f_\theta(a), Y_a) \tag{3.3}$$

**DeepFool**

The DeepFool algorithm proposed in [77] is an iterative algorithm that aims to minimize the perturbation size. Assuming the neural networks utilize Softmax as the activation function in the last layer, DeepFool keeps executing equation (3.4) until $a$ can be classified as Normal by $f_\theta$.

$$a = a - \frac{f_\theta(a)}{\|\nabla_a f_\theta(a)\|_2^2} \nabla_a f_\theta(a) \tag{3.4}$$

### 3.3.4 SearchFromFree Framework

#### Random Initialization

The state-of-the-art AML algorithms are originally designed for CV applications where the main constraint is the magnitude of the adversarial perturbation. However, as demonstrated in equation (3.1), the purpose of the attacker is to minimize the adversarial power consumption measurements reported to the utility to reduce his/her cost.

Different from CV applications where the input images are given and static, the energy thief can freely modify the smart meter's measurements. Since the AML algorithms can constrain the adversarial perturbation to be small, it is intuitive that the crafted adversarial measurements will be small if the initial measurements for the DL model are small. In practice, the minimum power consumption should be zero, which indicates a free electricity bill to the energy thief. However, a constant zero measurement vector will result in constant adversarial measurements, which is obviously abnormal to the utilities. We propose a scheme that randomly initializes adversarial measurements according to a Gaussian distribution $a \sim \mathcal{N}(0, \sigma^2)$ with the mean value set to zero ($\mu = 0$) and the standard deviation $\sigma$ set to a small value. We set all the non-negative values in $a$ to zero to meet constraint (3.1c). This initialization approach is compatible with different AML algorithms.

#### Step-size Searching Scheme

The iterative attacks usually have worse transferability. For example, the multiple-step DeepFool attack executes an iteration process and return the adversarial example as soon as it is misclassified by the given model. Empirically, the example is unique to the given model and may have low transferability. In energy theft detection, the adversarial measurements from the attacker are always smaller than normal measurements. Statistically, for a trained model, larger adversarial measurement vectors will have a higher probability to bypass the detection. Since the attacker's model $f'_\theta$ may share a similar manifold with $f_\theta$, we design a step-size

iterative scheme to search adversarial measurements that share the best transferability to increase the performance of black-box iterative attacks.

Enabled by the random initialization approach, our step-size scheme can be represented by Algorithm 1.

**Algorithm 1:** SearchFromFree Iteration Algorithm

```
1  Input: f'_θ', step, size, σ
2  Output: a
3  function ssf-Iter(f'_θ', step, size, σ)
4      initialize a ~ N(0, σ^2)
5      a = clip(a, min=0)
6      initialize stepNum = 0
7      while stepNum ≤ step − 1 do
8          calculate gradient G = ∇_a L(f'_θ'(a), Y_a)
9          r = G · size/max(abs(G))
10         update a = a + r
11         a = clip(a, min=0)
12         stepNum++
13     end
14     return a
15 end
```

The **ssf-Iter** function in Algorithm 1 has four inputs, including the local ML model $f'_{\theta'}$ and three positive constant parameters. The constant *step* limits the maximum number of iteration while *size* defines the maximum modification of $a$ in each iteration. As shown by Line 4, we empirically initialize $a$ according to a Gaussian distribution with the standard deviation value equals to $\sigma$ and mean value equals to zero. The iteration process gradually increases $\|a\|_1$ to have a higher probability to bypass the detection. Therefore, a small initial $a$ will finally lead to a smaller $\|a\|_1$ and the attacker can make more profit. The perturbation $r$ generated from the loss gradient may cause negative measurement values in $a$. We set all the negative values to zero to generate a feasible adversarial measurement vector $a$, as shown by Line 5 and 11.

## 3.4 Simulation Evaluation

### 3.4.1 Dataset Structure

We employ the smart meter data published by the Irish Social Science Data Archive (ISSDA) [6] as it is widely used as a benchmark for energy theft detection in related literature [118, 116, 78, 49, 85, 52]. The dataset contains the smart meter energy consumption measurement data of over 5000 customers in the Irish during 2009 and 2010. We assume all the measurement data in the dataset is normal since the customers agreed to install the smart meters and participated in the research project. There are missing and illegal measurements in the raw dataset and we pre-process the dataset by filtering out the incomplete measurements. We regulate the time-series measurement data into daily reading vectors and obtain the dataset $D_{daily}$. Since the power consumption measurements are recorded every 30 minutes, each daily reading measurement vector will contain 48 power consumption measurements.

To solve the shortage of real-world energy theft datasets, we employ the false measurement data generation approach proposed in [116] to simulate the energy theft measurements, which is a benchmark method used in previous literature [118, 31, 78, 49, 85]. [116] presents six energy theft scenarios, as shown in Table 3.1. The attack $h_1$ multiples the original reading with a constant while $h_2$ with a random constant vector generated from a uniform distribution. The $h_3$ considers that the energy thief reports zero consumption during a specific period. The $h_4$ scenario happens when an attacker constantly reports the mean consumption. $h_5$ is similar to $h_2$ but multiplying the random constant vector with the mean value instead of the real measurements. Finally, $h_6$ reverses the records of a day so that the small consumption will be reported during the periods in which the electricity price is lower.

A synthetic dataset is generated based $D_{daily}$. We randomly sample 180,000 daily records from $D_{daily}$ and modify half records in the sampled dataset according to the attack scenarios described in Table 3.1. We label all normal records as 0 and polluted records as 1 with One-hot encoding. We finally obtain the defender dataset $D_{defender} : \{M_{180,000\times48}, Y_{180,000\times1}\}$. We simulate the dataset $D_{attacker}$ for the attacker in the same way.

**Table 3.1:** Energy Theft Attack Scenarios [116]

| **Attack Scenario** |
|---|
| $h_1(m_t) = \alpha m_t,\ \alpha \sim Uniform(0.1, 0.8)$ |
| $h_2(m_t) = \beta_t m_t,\ \beta_t \sim Uniform(0.1, 0.8)$ |
| $h_3(m_t) = \begin{cases} 0 & \forall t \in [t_i, t_f] \\ m_t & \forall t \notin [t_i, t_f] \end{cases}$ |
| $h_4(m_t) = E(m)$ |
| $h_5(m_t) = \beta_t E(m)$ |
| $h_6(m_t) = m_{48-t}$ |

### 3.4.2 Model Training

The evaluation experiments are conducted based on three types of deep neural networks (DNN), feed-forward neural network (FNN), CNN, and RNN. We train three DL models for the defender (utilities) and three separate models for the attacker with $D_{defender}$ and $D_{attacker}$ respectively. For each model, 20% records in $D_{defender}$ or $D_{attacker}$ are randomly sampled for testing the rest 80% for training. We manually tuned the parameters of the model training and the performances of corresponding models are shown in Table 3.2. Overall, the RNNs achieve the best classification performance since they have an intrinsic advantage in learning the pattern of time-series data. The structures of the neural networks are shown in Table 3.3. All the DNNs are implemented with the TensorFlow and Keras library. The training process is conducted on a Windows 10 PC with an Intel Core i7 CPU, 16 GB memory, and an NVIDIA GeForce GTX 1070 graphic card to accelerate the training process. The models are optimized with a Rmsprop optimizer.

### 3.4.3 Metrics and Baselines

#### Metrics

We set two metrics to evaluate the performance of adversarial attacks. Since all the test records are false measurements (generated by our random initialization scheme), the first metric is the detection recall (TP/(TP + FN)) of the defender's models under adversarial attacks.

We set two metrics to evaluate the performance of adversarial attacks. Since all the test records are false measurements (generated by our random initialization scheme), the first metric is the detection recall (TP/(TP + FN)) of the defender's models under adversarial attacks. Meanwhile, it is straightforward that a larger adversarial measurement vector will have a higher probability to bypass the detection. Therefore, we set the average $L_1$-Norm of the adversarial measurement vectors as the second evaluation metric. In our experiment, the average $L_1$-Norm of all normal measurement records is 32.05 kWh.

**Table 3.2:** Model Performance

| **Model** | **Accuracy** | **False Positive Rate** |
|---|---|---|
| $f_{FNN}$ | 86.9% | 10.01% |
| $f'_{FNN}$ | 86.87% | 14.01% |
| $f_{RNN}$ | 97.5% | 2.58% |
| $f'_{RNN}$ | 97.48% | 2.62% |
| $f_{CNN}$ | 93.49% | 7.79% |
| $f'_{CNN}$ | 93.28% | 6.41% |

**Table 3.3:** Model Structures

| **Networks** | **FNN** | | **RNN** | | **CNN** | |
|---|---|---|---|---|---|---|
| **Models** | $f_{FNN}$ | $f'_{FNN}$ | $f_{RNN}$ | $f'_{RNN}$ | $f_{CNN}$ | $f'_{CNN}$ |
| **Layer 0** | input 48 | input 48 | input $48 \times 1$ | input $48 \times 1$ | input $6 \times 8$ | input $6 \times 8$ |
| **Layer 1** | 128 Dense | 168 Dense | 256 LSTM | 246 LSTM | 128 Conv2D | 156 Conv2D |
| **Layer 2** | 256 Dense | 328 Dense | Dropout 0.25 | Dropout 0.25 | 128 Conv2D | 214 Conv2D |
| **Layer 3** | 128 Dense | 168 Dense | 168 LSTM | 148 LSTM | MaxPooling2D | MaxPooling2D |
| **Layer 4** | Dropout 0.25 | 128 Dense | Dropout 0.25 | Dropout 0.25 | Dropput 0.25 | Dropput 0.25 |
| **Layer 5** | 32 Dense | Dropout 0.25 | 128 LSTM | 108 LSTM | flatten | flatten |
| **Layer 6** | Dropout 0.25 | 64 Dense | 2 Dense Softmax | 2 Dense Softmax | 32 Dense | 48 Dense |
| **Layer 7** | 2 Dense Softmax | Dropout 0.25 | - | - | Dense 2 Softmax | Dense 2 Softmax |
| **Layer 8** | - | 2 Dense Softmax | - | - | - | - |

The models $f_*$ act as the defenders while $f'_*$ as attackers. The activation function of each layer is $ReLu$ unless specifically noted. The kernel size is $3 \times 3$ for CNN models.

#### Baselines

We set up two **vanilla black-box attackers** as baselines to demonstrate the effectiveness of adversarial attacks. The first vanilla attacker **VA1** will gradually try different $\alpha$ of $h_1$ as defined in Table 3.1 while the second vanilla attacker **VA2** generates uniformly distributed measurement vector between 0 and a variable $u$.

### 3.4.4 Experimental Result

The evaluation experiments are conducted with 1,000 adversarial measurement vectors. All the DNN's detection recall of the original randomly initialized adversarial measurement vectors is 100%. The standard deviation of the Gaussian distribution used for initialization is set to $\sigma = 0.0001$.

#### Vanilla Attacks

As expected, the detection recall of the defenders' models decreases with the parameter $\alpha$ increases under **VA1** attack. This indicates that **VA1** has a higher success probability if he/she was willing to decrease his/her stolen profit. From Fig. 3.1, if **VA1** wants to have a relatively high success probability for energy theft, such as over 65%, the required power consumption bill should be over 20 kWh ($\alpha > 0.65$).

As shown in Fig. 3.2, the detection recall of RNN and CNN remains high (over 95%) with the parameter $u$ increases. This indicates that a uniformly distributed consumption measurement vector is obviously abnormal for models that are trained to learn the daily electricity consumption patterns. Overall, the **VA2** attack is not effective for energy theft.

#### State-of-the-art Approaches

We apply the random initialization approach to the state-of-the-art AML algorithms and evaluate the attack performances under the white-box and black-box settings. Similar to Algorithm 1, we map all the negative values in the adversarial measurements to zero to be feasible. We test different $\epsilon$ values for FGSM and FGV, and evaluation result is shown in Fig. 3.3 and Fig. 3.4 respectively.

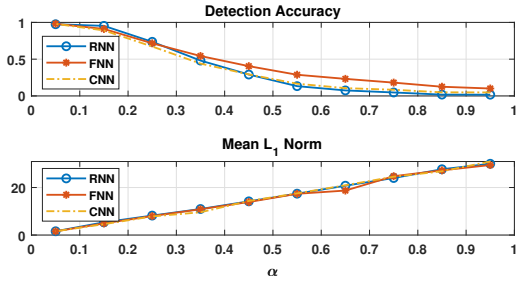


**Figure 3.1:** Vanilla attacker 1 evaluation result.

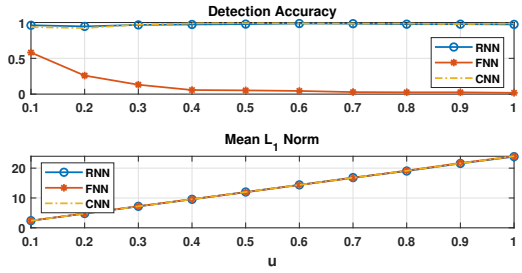


**Figure 3.2:** Vanilla attacker 2 evaluation result.

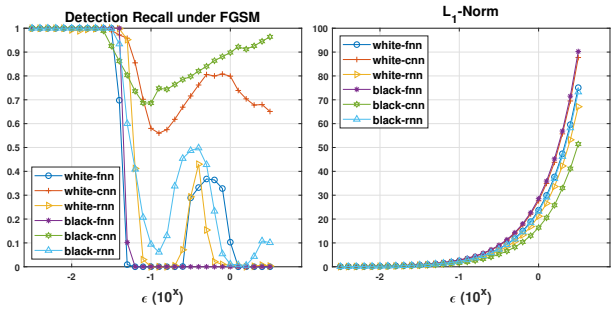


**Figure 3.3:** FGSM Evaluation Result

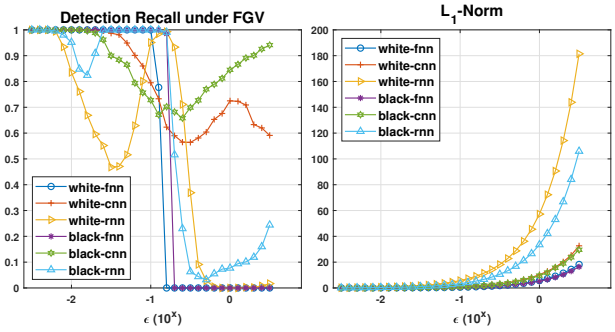


**Figure 3.4:** FGV Evaluation Result

From Fig. 3.3, we can learn that FGSM can achieve notable attack performance for FNN and RNN. In the black-box settings, the probability of bypassing RNN detection is over 90% while the adversarial measurement's $L_1$-Norm is only 2.9 kWh ($\epsilon = 10^{-0.9}$). The attack performance is even better for FNN. When $\epsilon = 10^{-1.2}$, the energy thief obtains a 100% detection bypassing rate with a 1.4 kWh electricity bill. The single-step attack FGSM does not perform well for CNN detection. The best evasion rate is around 44% for the white-box attack and 32% for the black-box attack.

Overall, the attack performance of FGV is slightly worse than FGSM in black-box settings but is still effective, as shown in Fig. 3.4. For example, the black-box attack to RNN obtains a 94% detection bypassing rate while the $L_1$-Norm is 10.6 kWh ($\epsilon = 10^{-0.5}$), which is higher than FGSM (2.9 kWh) but is still smaller than the normal measurements (32.05 kWh). Similar to FGSM, the FGV achieves the best performance for FNN detection, followed by RNN and CNN.

The evaluation result of the iterative attack DeepFool is summarized in Table 3.4. The iterative attack demonstrates notable performances in white-box settings. The detection recall of all three DNNs becomes 0% under white-box attacks while the $L_1$-Norm is smaller than 1 kWh. However, as expected, the adversarial measurements generated by iterative attacks are less transferable. Under the black-box setting, the DeepFool attack only shows effectiveness in FNN detection while the detection recall of CNN and RNN remains 100%.

#### SearchFromFree Iteration Algorithm

We then evaluate the performance of our **ssf-Iter** algorithm, an iterative attack algorithm that utilizes a step-size scheme to search for transferable adversarial measurements, as shown in Fig. 3.5 and Fig. 3.6.

From Fig. 3.5, we can learn that our algorithm performs best in FNN, followed by CNN and RNN. In most cases, the detection recall of three DNNs approaches to zero under the white-box attack while the adversarial measurements are still small enough (around 1 kWh).

**Table 3.4:** DeepFool Evaluation Performance

| **Model** | **FNN** | | **CNN** | | **RNN** | |
|---|---|---|---|---|---|---|
| **Metric** | recall | size | recall | size | recall | size |
| **white-box** | 0% | 0.94 | 0% | 0.23 | 0% | 0.02 |
| **black-box** | 17.4% | 1.14 | 100% | 0.115 | 100% | 0.06 |

$*$ 'size' is the $L_1$-Norm of adversarial measurements (kWh)

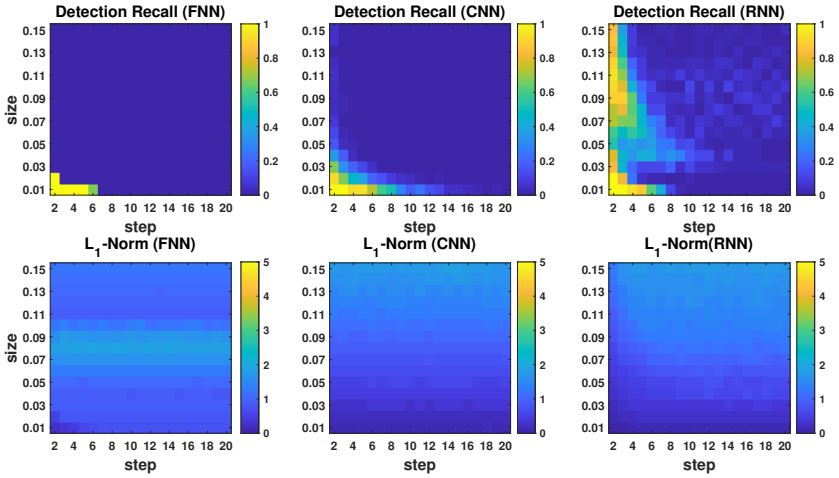


**Figure 3.5:** White-box SearchFromFree

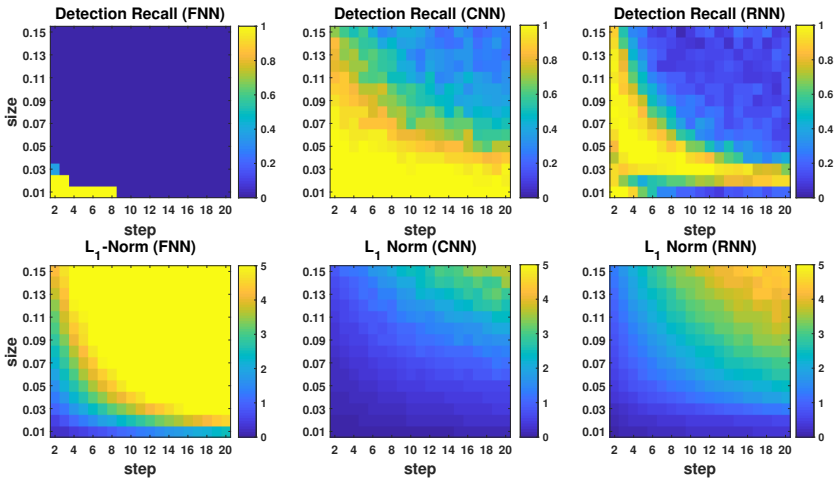


**Figure 3.6:** Black-box SearchFromFree.

As expected, the adversarial performances under the black-box setting are worse than the white-box setting, as shown in Fig. 3.6. In general, the probability of bypassing the detection is lower and the corresponding measurement size is larger. The attacker is required to pay a higher cost ($L_1$-Norm of adversarial measurements) in the black-box settings to obtain the same detection bypassing rate in the white-box settings. Statistically, the FNN detection still performs worst under black-box adversarial attacks. By analyzing the corresponding evaluation parameters, we can learn that the attacker can bypass the FNN's detection with nearly 100% success probability while the average $L_1$-Norm is around 1 kWh. For CNN detection, our adversarial attack can achieve over 70% successful rate while keeping the $L_1$-Norm below 4 kWh.

Attack performance is better for RNN detection. In most attack scenarios, the RNN's detection recall is below 30% while the $L_1$-Norm is lower than 3 kWh. It is worth noting that if the attacker sets *size* to 0.01, the adversarial attack can obtain over 80% successful probability with an around 0.2 kWh measurement size. Compared with the DeepFool attack, our algorithm achieves similar performance in the white-box settings and better transferability under the black-box settings.

**Parameter Selection:** Fig. 3.5 and Fig. 3.6 show that the attack performances can be impacted by the parameters in Algorithm 1. However, from the 2D pixel figures, we can observe that the attack performances follow specific patterns according to the two parameters. Overall, as long as the parameters fall in a specific range, the attack performance will be satisfied. Meanwhile, by comparing Fig. 3.5 and Fig. 3.6, we can learn that the performances of black-box attacks share similar manifolds with white-box attacks under our step-size scheme. This indicates that the attacker can select the algorithm parameters based on the performances of his/her local DL models. In practical scenarios, different attackers may also collude together to search for the parameters that produce the best attack performance.

# Chapter 4

# Control Domain Adversarial Attacks in CPS: ConAML

In this section, we study the adversarial attacks in control domain CPSs by proposing the ConAML framework.

## 4.1 Introduction

Machine learning (ML) has shown promising performance in many real-world applications, such as image classification [43], speech recognition [36], and malware detection [114]. In recent years, motivated by the promotion of cutting-edge communication and computational technologies, there is a trend to adopt ML in various control domain cyber-physical system (CPS) applications, such as data center thermal management [61], agriculture ecosystem management [23], power grid attack detection [83], and industrial control system anomaly detection [53].

However, recent research has demonstrated that the superficially well-trained ML models are highly vulnerable to adversarial examples [96, 35, 89, 75, 55, 25, 76]. In particular, adversarial machine learning (AML) technologies enable attackers to deceive ML models with well-crafted adversarial examples by adding small perturbations to legitimate inputs. As CPSs have become synonymous to security-critical infrastructures such as the power grid,

nuclear systems, avionics, and transportation systems, such vulnerabilities can be exploited leading to devastating consequences.

AML research has received considerable attention in artificial intelligence (AI) communities and it mainly focuses on computational applications such as computer vision. However, it is not applicable to control domain CPSs because the inherent properties of CPSs render the widely-used threat models and AML algorithms in previous research infeasible. In general, the existing AML research makes common assumptions on the attacker's knowledge and the adversarial examples. The attacker is assumed to have full knowledge of the ML inputs and these features are assumed to be mutually independent. For example, in computer vision AML [35], the attacker is assumed to know all the values of pixels of an image and there is no strict dependency among the pixels. However, this is not realistic for attacks targeting control domain CPSs. CPSs are usually large and complex systems whose data sources are heterogeneous and geographically distributed. The attacker may compromise a subset of sensors and modify their measurement data. Generally, for the uncompromised data sources, the attacker cannot even know the measurements, let alone making modifications. Furthermore, for robustness and resilience reasons, control domain CPSs usually employ redundant data sources and incorporate faulty data detection mechanisms. For example, in the power grid, redundant phasor measurement units (PMUs) are deployed in the field to measure frequency and phase angle, and residue-based bad data detection is employed to detect and recover from faulty data for state estimation [106]. Therefore, the features of ML applications in CPS are not only dependent but also subject to the physical constraints of the system. A simple example of constraints is shown in Figure 4.1. All three meters are measuring the electric current (Ampere) data. If an attacker compromises *Meter1*, *Meter2*, and *Meter3*, no matter what modification the attacker makes to the measurements, the compromised measurement of *Meter1* should always be the sum of that of *Meter2* and *Meter3* due to Kirchhoff's laws. Otherwise, the crafted measurements will be detected by the bad data detection mechanism and obviously anomalous to the power system operators. In addition to distributed data sources and physical constraints, sensors in real-world CPSs are generally configured to collect data with a specific sampling rate. A valid adversarial attack needs to be finished within the CPS' sampling period.

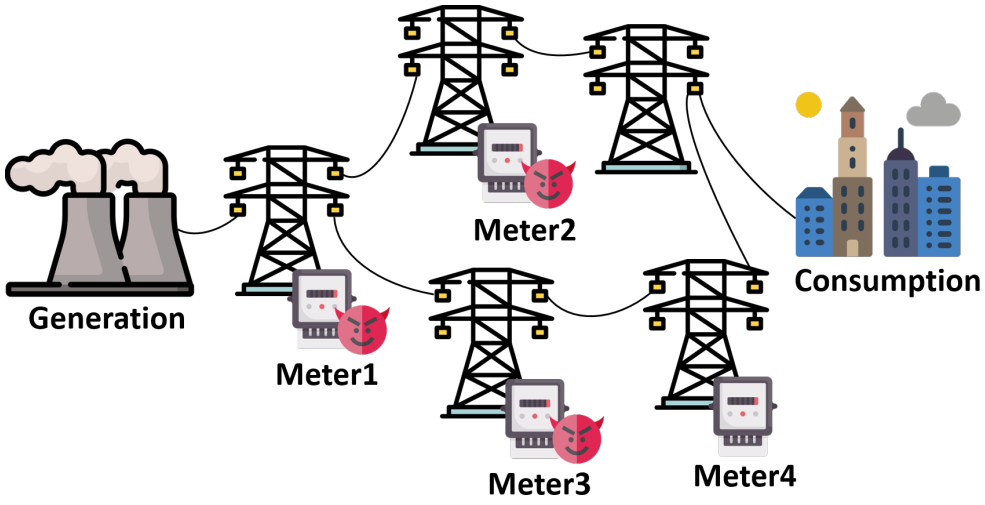


**Figure 4.1:** A CPS example (power grids).

The intrinsic properties of CPS pose stringent requirements for the attackers. The attacker is now required to overcome:

- **Model constraint:** No access to the original CPS DNNs.
- **Sensor constraint:** Can only compromise a portion of sensors and modify their values.
- **Knowledge constraint:** No access to the ML models and the measurement values of uncompromised sensors.
- **Physical constraint:** The adversarial examples need to meet the physical constraints defined by the system.
- **Time constraint:** Attacks needs to be completed within a sample period of the sensors.

to launch an effective attack that deceives the ML applications deployed in CPSs. However, in this chapter, we show that the ML applications in CPSs are susceptible to handcrafted adversarial examples even though such systems naturally pose a greater barrier for the attacker.

In this chapter, we propose constrained adversarial machine learning (**ConAML**), a general AML framework that incorporates the above constraints of CPSs. We firstly design a universal adversarial measurement algorithm to solve the knowledge constraint. After that, without loss of generality, we present a practical best-effort search algorithm to effectively generate adversarial examples under linear physical constraints which are one of the most common constraints in real-world CPS applications. Meanwhile, we set the maximum iteration number to control the time cost of the attack. We implement our algorithms with ML models used in three CPS applications and mainly focus on neural networks due to its transferability. Our main contributions are summarized as follows:

- We highlight the potential vulnerability of deploying ML in CPSs, analyze the different requirements for AML applied in CPSs with regard to the general computational applications, and present a practical threat model for AML in CPSs.

- We formulate the mathematical model of ConAML by incorporating the physical constraints of the underlying system. To the best of our knowledge, this is also the first work that investigates the physical mutual dependency among the ML features in AML research.

- We proposed ConAML, an AML framework that contains a series of AML algorithms to generate adversarial examples under the corresponding constraints.

- We assess our algorithms with three typical CPS applications, including incident detection in transportation system, FDIA detection in the power grids and anomaly detection water treatment system, where ML are intensively investigated for attack detection in the research literature [40, 62, 115, 87, 86, 120, 83, 111, 44, 47, 10, 81, 105, 45, 53, 28, 18, 27, 5]. The evaluation results show that the adversarial examples generated by our algorithms can effectively bypass the ML-powered detection systems in the three CPSs.

## 4.2 Related Work

AML of neural networks was discovered by Szegedy *et al.* [96] in 2013. They found that a deep neural network used for image classification can be easily fooled by adding a certain, hardly perceptible perturbation to the legitimate input. Moreover, the same perturbation can cause a different network to misclassify the same input even when the network has a different structure and is trained with a different dataset, which is referred to as the transferability property of adversarial examples in the following research. After that, in 2015, Goodfellow *et al.* [35] proposed the Fast Gradient Sign Method (FGSM), an efficient algorithm to generate adversarial examples. The Fast Gradient Value (FGV) method proposed by Rozsa *et al.* [89] is a simple variant of FGSM, in which the authors utilize the raw gradient instead of the sign. In 2016, Moosavi-Dezfooli *et al.* presented *DeepFool* which searches for the closest distance between the original input to the decision boundary in high dimensional data space and iteratively builds the adversarial examples [75]. According to [55], single-step attack methods have better transferability but can be easily defended. Therefore, multi-steps

methods, such as iterative methods [55] and momentum-based methods [25], are presented to enhance the effectiveness of attacks. The above methods generate individual adversarial examples for each legitimate inputs. In 2017, Moosavi-Dezfooli *et al.* designed universal adversarial perturbations to generate perturbations regardless of the ML model inputs [76].

Research on AML applications continues growing rapidly. Sharif *et al.* generated adversarial examples to attack a state-of-the-art face-recognition system and achieved a notable result [92]. Grossee *et al.* constructed an effective attack that generated adversarial examples against Android malware detection models [37]. The adversarial attacks that target real-world applications also increase. In 2014, Laskov *et al.* developed a taxonomy for practical adversarial attacks based on the attackers' capability and launched evasion attacks to *PDFRATE*, a real-world online machine learning system to detect malicious PDF malware [88]. Followed by Xu *et al.* , in 2016, they utilized a genetic programming algorithm to generate evasion adversarial examples to evaluate the robustness of ML classifiers [110]. Their methods were evaluated with *PDFRATE* and *Hidost*, another PDF malware classifier. In 2018, Li *et al.* presented *TEXTBUGGER*, a framework to effectively generate adversarial text against deep learning-based text understanding (DLTU) systems and achieved state-of-the-art attack performance [58].

In addition to pure computation and cyberspace attacks, AML techniques that involve the physical domain are drawing more and more attention. Kurakin *et al.* presented that ML models are still vulnerable to adversarial examples in physical world scenarios by feeding a phone camera captured adversarial image to an ImageNet classifier [54]. In 2016, Carlini *et al.* presented hidden voice commands and demonstrated that well-crafted voice commands which are unintelligible to human listeners, can be interpreted as commands by voice controllable systems [15]. [99] and [67] investigated the security of machine learning models used in autonomous driving cars. In 2018, [32] showed that an attacker can generate adversarial examples by modifying a portion of measurements in CPSs, and presented an anomaly detection model where each sensor's reading is predicted as a function of other sensors' readings. After that, Erba *et al.* also studied the AML in CPS and consider the physical constraints [26]. They employed an autoencoder that trained on normal system data to reconstruct the bad inputs to match the physical behavior. However, both [32] and [26]

allow the attacker to know all the measurements and the generated adversarial examples may still violate the physical constraints.

More related work on adversarial examples, including the generation algorithms and related applications, can be found in [113].

## 4.3 System and Threat Model

### 4.3.1 ML-Assisted CPSs

Generally, a CPS can be simplified as a system that consists of four parts, namely sensors, actuators, the communication network, and the control center [18], as shown in Figure 4.2. The sensors measure and quantify the data from the physical environment, and send the measurement data to the control center through the communication network. In practice, the raw measurement data will be filtered and processed by the gateway according to the error checking mechanism whose rules are defined by human experts based on the properties of the physical system. Measurement data that violates the physically defined rules will be removed.

Similar to [26], we consider the scenario that the control center utilizes ML model(s) to make decisions (classification) based on the filtered measurement data from the gateway directly, and the features used to train the ML models are the measurements of sensors respectively. The target of the attacker will be deceiving the ML model(s) in CPSs to output wrong (classification) results without being detected by the gateway by adding perturbations to the measurements of the compromised sensors.

### 4.3.2 Threat Model

Adversarial attacks can be classified according to the attacker's capability and attack goals [88, 113, 16]. In this work, we consider the integrity attack that the attacker generates adversarial perturbations to the ML inputs to deceive the ML model to make incorrect classification outputs.

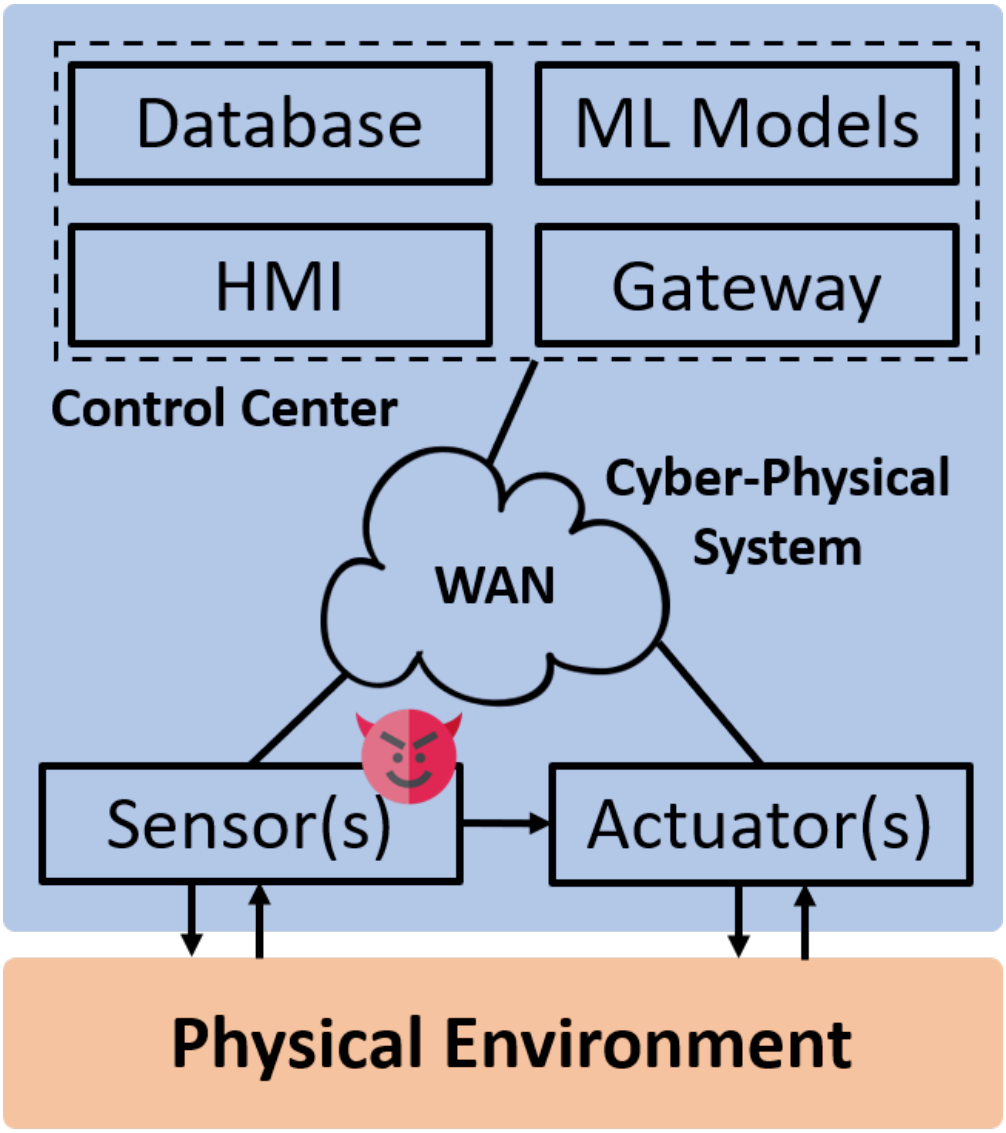


**Figure 4.2:** Machine learning-assisted CPS architecture.

There are several inherent properties of CPS that pose specific requirements for adversarial attacks. First, in CPS, ML models are usually placed in the control centers and other centralized locations which employ comprehensive and advanced security measures such as air-gapped networks. It is highly unlikely for the attacker to have access to the models and a black-box attack should be considered. Second, we assume that the attacker cannot access the training dataset for the same reason as above, but has access to an alternative dataset such as historical data that follows a similar distribution to train their models. It is possible for the attacker to obtain historical data in practice, for instance, temperature data for load forecasting, earthquake sensor data, flood water flow data, and traffic flow data, since these data are usually published or shared among multiple parties.

To launch adversarial attacks, the attacker is assumed to compromise a certain number of sensors, and can freely eavesdrop and modify their measurement data. These sensors are deployed in the wild and their security is hard to guarantee. In real attack scenarios, this can be implemented by either directly compromising the sensors, such as device intrusion or attacking the communication network, such as man-in-the-middle attacks. However, due to the vastly distributed nature of sensors in CPS, it is only reasonable for the attacker to compromise a subset of the data sources but not all of them. For the uncompromised sensors, the attacker can neither know their measurement values nor make modifications. This constraint indicates that the attacker has limited knowledge of the ML inputs.

Meanwhile, the attacker is further required to generate adversarial examples that meet the constraints imposed by the physical laws and system topology and evade any built-in detection mechanisms in the system. Specifically, since they are very common in real-world CPSs, we will mainly focus on linear constraints in this work, including both linear equality constraints and linear inequality constraints. An example of the linear inequality constraint is shown in Figure 4.3. All the meters in Figure 4.3 are measuring water flow which follows the arrows' direction. If an attacker wants to defraud the anomaly detection ML model of a water treatment system by modifying the meters' readings, the adversarial measurement of *Meter1* should always be larger than the sum of *Meter2* and *Meter3* due to the physical structure of the pipelines. Otherwise, the poisoned inputs will be obviously anomalous to the victim (system operator) and detected automatically by the error checking mechanisms.

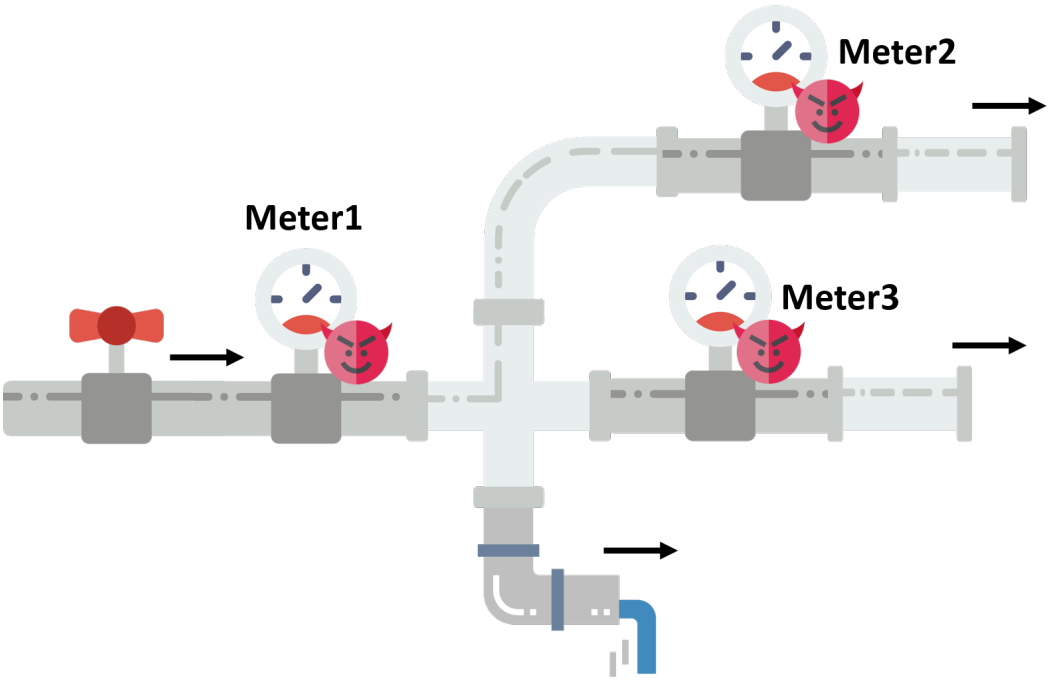


**Figure 4.3:** A CPS example (water pipelines).

In practice, many of the linear constraints can be explicitly abstracted by the attacker from the compromised measurement data. Meanwhile, the practical CPSs usually have built-in tolerance for noise and normal fluctuation in the measurements so that the approximately estimated constraints will still be effective for the adversarial attackers. Therefore, we assume that the attacker know the linear constraints among the compromised measurements.

The real-world CPSs, such as the Supervisory Control and Data Acquisition (SCADA), will have a constant measurement sampling rate (frequency) configured for their sensors. The attacker who targets CPSs' ML applications is then required to generate a valid adversarial example within a measurement sampling period.

We summarize the threat model as follows:

- We assume the attacker has no access to the system operator's trained model in the control center, including the hyper-parameters and the related dataset. However, the attacker has an alternative dataset as an approximation of the defender's (system operator's) training dataset to train his/her ML models.
- The attacker can compromise a subset of sensors in the CPS and make modifications to their measurement data. However, the attack can neither know nor modify the measurements of uncompromised sensors.
- The attacker can know the linear constraints of the measurements imposed by the physical system.

### 4.3.3 Physical Constraint Mathematical Representation

In this subsection, we present the mathematical definition of the physical linear constraints of the ML inputs and represent the AML as a constrained optimization problem.

#### Notations

To simplify the mathematical representation, we will use $A_B = \left[a_{b_0}, a_{b_1}, ..., a_{b_{n-1}}\right]$ to denote a sampled vector of $A = [a_0, a_1, ..., a_{m-1}]$ according to $B$, where $B = [b_0, b_1, ..., b_{n-1}]$ is a vector of sampling index. For example, if $A = [a, b, c, d, e]$ and $B = [0, 2, 4]$, we have $A_B = [a, c, e]$.

We assume there are totally $d$ sensors in a CPS, and each sensor's measurement is a feature of the ML model $f_\theta$ in the control center. We use $S = [s_0, s_1, ..., s_{d-1}]^T$ and $M = [m_0, m_1, ..., m_{d-1}]^T$ to denote all the sensors and their measurements respectively. The attacker compromised $r$ sensors in the CPS and $C = [c_0, c_1, ..., c_{r-1}]$ denotes the index vector of the compromised sensors. Obviously, we have $\|C\| = r$ and $0 < r \leq d$. Meanwhile, the uncompromised sensors' indexes are denoted as $U = [u_0, u_1, ..., u_{d-r-1}]$ ($\|U\| = d - r$).

$\Delta = [\delta_0, \delta_1, ..., \delta_{d-1}]^T$ is the adversarial perturbation to be added to $M$. However, the attacker can only inject $\Delta_C = \left[\delta_{c_0}, \delta_{c_1}, ..., \delta_{c_{r-1}}\right]^T$ to $M_C$ while $\Delta_U = 0$. The polluted adversarial measurements become $M_C^* = M_C + \Delta_C$, and $m_{c_i}^* = m_{c_i} + \delta_{c_i}$ ($0 \leq i \leq r-1$). Apparently, we have $\delta_i = \delta_{c_j}$ when $i = c_j$, $i \in C$, and $\delta_i = 0$ when $i \notin C$. Similarly, the crafted adversarial example $M^* = \left[m_0^*, m_1^*, ..., m_{d-1}^*\right] = M + \Delta$ is fed into $f_\theta$. We have $m_i^* = m_{c_j}^*$ when $i = c_j$, $i \in C$ and $m_i^* = m_i$ when $i \notin C$. All the notations are summarized in Table 4.1.

### Mathematical Presentation

For **linear equality constraints**, such as the current measurements (Amperes) of the three meters in Figure 4.1, we suppose there are $k$ constraints of the compromised measurements $M_C$ that the attacker needs to meet, and the $k$ constraints can be represented as follow:

$$\begin{cases} \phi_{0,0} \cdot m_{c_0} + ... + \phi_{0,r-1} \cdot m_{c_{r-1}} = \phi_{0,r} \\ \phi_{1,0} \cdot m_{c_0} + ... + \phi_{1,r-1} \cdot m_{c_{r-1}} = \phi_{1,r} \\ ... \\ \phi_{k-1,0} \cdot m_{c_0} + ... + \phi_{k-1,r-1} \cdot m_{c_{r-1}} = \phi_{k-1,r} \end{cases} \tag{4.1}$$

The above constraints can be represented as (4.2). We have $\Phi_{k\times r} = [\Phi_0, \Phi_1, ..., \Phi_{k-1}]^T$, where $\Phi_i = [\phi_{i,0}, \phi_{i,1}, ..., \phi_{i,r-1}]$ ($0 \leq i \leq k-1$), $\Phi_{i,j} = \phi_{i,j}$ ($0 \leq i \leq k-1, 0 \leq j \leq r-1$) and $\tilde{\Phi} = [\phi_{0,r}, \phi_{1,r}, ..., \phi_{k-1,r}]^T$.

$$\Phi_{k\times r} M_C = \tilde{\Phi} \tag{4.2}$$

**Table 4.1:** List of Notations

| Symbol | Description |
|---|---|
| $f_\theta$ | The trained model with hyperparameter $\theta$ |
| $S$ | The vector of sensors |
| $M$ | The vector of measurements of $S$ |
| $\Delta$ | The perturbations vector added to $M$ |
| $M^*$ | The sum of $\Delta$ and $M$. The vector of compromised input |
| $C$ | The vector of the indexes of compromised sensors or measurements |
| $U$ | The vector of the indexes of uncompromised sensors or measurements |
| $Y$ | The original class of the measurement $M$ |
| $Y'$ | The target class of the measurement $M^*$ |
| $\Phi$ | The linear constraint matrix |

The attacker generates the perturbation vector $\Delta_C$ and adds it to $M_C$ such that $f_\theta$ will predict the different output. Meanwhile, the crafted measurements $M_C^* = \Delta_C + M_C$ should also meet the constraints in (4.2) to avoid being noticed by the system operator or detected by the error checking mechanism.

Formally, the attacker who launches AML attacks needs to solve the following optimization problem:

$$\max_{\Delta_C} \quad L(f_\theta(M^*), Y) \tag{4.3a}$$

$$s.t. \quad M_C^* = M_C + \Delta_C \tag{4.3b}$$

$$\Phi_{k\times r} M_C = \tilde{\Phi} \tag{4.3c}$$

$$\Phi_{k\times r} M_C^* = \tilde{\Phi} \tag{4.3d}$$

$$M^* = M + \Delta \tag{4.3e}$$

$$\Delta_U = 0 \tag{4.3f}$$

where $L$ is a loss function, and Y is the original class label of the input vector $M$.

In addition, the **linear inequality constraints** among the compromised measurements can be represented as equation (4.4), and the constrained optimization problem to be solved is also similar to (4.3) but replacing (4.3c) with $\Phi_{k\times r} M_C \leq \tilde{\Phi}$ and (4.3d) with $\Phi_{k\times r} M_C^* \leq \tilde{\Phi}$ respectively.

$$\Phi_{k\times r} M_C \leq \tilde{\Phi} \tag{4.4}$$

## 4.4 Design of ConAML

The universal adversarial measurements algorithm is proposed in subsection 4.4.1 to solve the knowledge constraint of the attacker. Subsection 4.4.2 and subsection 4.4.4 analyze the

properties of physical linear equality constraints and linear inequality constraints in AML respectively and present the adversarial algorithms.

### 4.4.1 Universal Adversarial Measurements

**Algorithm 2:** Universal Adv-Measur Algorithm

**Input:** $f_\theta$, $MU$, $M_C$, $\lambda$, $Y$, $MaxItera$
**Output:** $M^*$
**function** uniAdvMeasur($f_\theta, MU, M_C, \lambda, Y, MaxItera$)
    initialize $\Delta = 0$
    build set $MUC = \{M_{C|U_0}, M_{C|U_1}, ..., M_{C|U_N}\}$
    set counter $cycNum = 0$
    **while** $cycNum < MaxItera$ **do**
        set $flag$ to 0
        **for** $M_{C|U_i}$ *in* $MUC$ **do**
            $\Delta =$ **onePerturGenAlgorithm**($\Delta, M_{C|U_i}$)
            **if** ***sampleEva***($f_\theta, Y, MUC, \Delta$) $< \lambda$ **then**
                set $flag$ to 1
                **break**
            **end**
        **end**
        **if** $flag$ *equals* 1 **then**
            **break**
        **end**
        $cycNum$++
    **end**
    **return** $M^* = M + \Delta$
**end**

We first deal with the challenge of the attacker's limited knowledge on the uncompromised measurements $M_U$. This challenge is difficult to tackle since the complete measurement vector $M$ is needed to obtain the gradient values in many AML algorithms [35, 89, 75, 55, 76] . In 2017, Moosavi-Dezfooli *et al.* proposed the universal adversarial perturbation scheme which generates image-agnostic adversarial perturbation vectors [76]. The identical universal adversarial perturbation vector can cause different images to be misclassified by the state-of-the-art ML-based image classifiers with high probability. The basic philosophy of [76] is to

iteratively and incrementally build a perturbation vector that can misclassify a set of images sampled from the whole dataset.

**Algorithm 3:** Sample Evaluation

```
1 Input: f_θ, Y, MUC, Δ
2 Output: Classification Accuracy
3 function sampleEva(f_θ, Y, MUC, Δ)
4 |   add perturbation Δ to all vectors in MUC
5 |   evaluate MUC with f_θ and label Y
6 |   return the classification accuracy of f_θ(MUC)
7 end
```

Inspired by their approach, we now present our universal adversarial measurements algorithm. We define an ordered set of $N$ sampled uncompromised measurements $MU = \{M_{U_0}, M_{U_1}, ..., M_{U_{N-1}}\}$, and use $M_{C|U_i}$ to denote the crafted measurement vector from $M_C$ and the sampled uncompromised measurement vector $M_{U_i}$. Here, $M_{C|U_i}$ is a crafted measurement vector with $\left\|M_{C|U_i}\right\| = d$. The uncompromised measurement vectors in $MU$ can be randomly selected from the attacker's alternative dataset.

Algorithm 2 describes a high-level approach to generate adversarial perturbations regardless of uncompromised measurements. The algorithm first builds a set of crafted measurement vector $MUC$ based on $MU$ and $M_C$, and then starts an iteration over $MUC$. The iteration process is limited to $MaxItera$ times to control the maximum time cost. The purpose is to find a universal $\Delta$ that can cause a portion of the vectors in $MUC$ misclassified by $f_\theta$. The function **sampleEva** described in Algorithm 3 evaluates $MUC$ and $Y$ with the ML model $f_\theta$ and returns the classification accuracy. $\lambda \in (0, 1]$ is a constant chosen by the attacker to determine the attack's success rate in $MUC$ according to $\Delta$. During each searching iteration, algorithm 2 builds and maintains the perturbation $\Delta$ increasingly using an adversarial perturbation generation algorithms, as shown by Line 10 in Algorithm 2. We will propose our methods to handle this problem in the next subsections.

Figure 4.4 presents a simple illustration of the iteration process in Algorithm 1. We assume there are three sensors' measurements $M = [m_0, m_1, m_2]$ and only one sensor's measurement $m_0 = \alpha$ is compromised by the attacker. The yellow, green and orange shallow areas in the plane $M_0 = \alpha$ represent the possible adversarial examples of the crafted

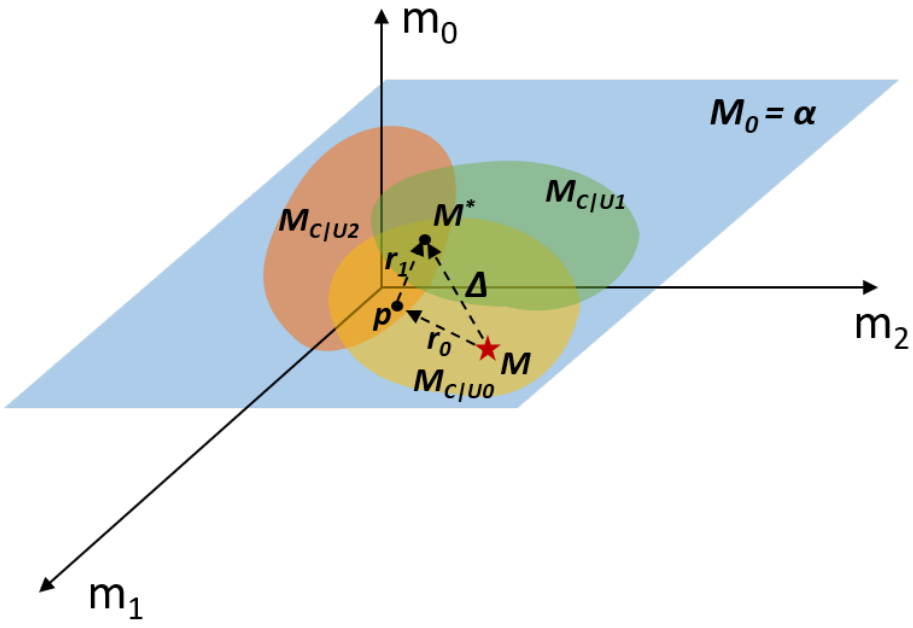


**Figure 4.4:** Iteration illustration.

measurement vector $M_{C|U_0}$, $M_{C|U_1}$, and $M_{C|U_2}$, respectively, where $U_i$ are randomly sampled measurements of uncompromised sensors ($m_1$ and $m_2$). The initial point $M$ (red ★) iterates twice ($r_0$ and $r_1$) and finally reaches $M^*$ with the universal perturbation vector $\Delta$. Therefore, $M^*$ is a valid adversarial example for all $M_{C|U_i}(i \in \{0, 1, 2\})$.

**Comparison of Methods:** Our approach is different from [76] in several aspects. First, the approach proposed in [76] has identical adversarial perturbations for different ML inputs while our approach actually generates distinct perturbations for each $M$. Second, the approach in [76] builds universal perturbations regardless of the real-time ML inputs. However, as the attacker has already compromised a portion of measurements, it is more effective to take advantage of the obtained knowledge. In other words, our perturbations are '**universal**' for $M_U$ but '**distinct**' for $M$. Finally, the intrinsic properties of CPSs require the attacker to generate a valid adversarial example within a sampling period while there is no enforced limitation of the iteration time in [76].

### 4.4.2 Linear Equality Constraints Analysis

As shown in [35] and [89], the fundamental philosophy of AML can be represented as (4.5).

$$M^* = M + \Delta = M + \epsilon \nabla_M L(f_\theta(M), Y) \tag{4.5}$$

However, directly following the gradient will not guarantee the adversarial examples meet the constraints in (4.2) and (4.4). With the constraints imposed by the physical system, the attacker is no longer able to freely add perturbation to original input using the raw gradient of the input vector. In this subsection, we will analyze how the linear equality constraints will affect the way to generate perturbation and use a simple example for illustration.

Under the threat model proposed in Section 4.3.2, the constraint of (4.3c) is always met due to the properties of the physical systems. We then consider the constraint (4.3d).

**Theorem 4.1.** *The sufficient and necessary condition to meet constraint (4.3d) is* $\Phi_{k\times r}\Delta_C = 0$.

*Proof.* If we replace $M_C^*$ in equation (4.3d) with equation (4.3b), we can get $\Phi_{k\times r}M_C^* = \Phi_{k\times r}(M_C+\Delta_C) = \Phi_{k\times r}M_C+\Phi_{k\times r}\Delta_C = \tilde{\Phi}$. From equation (3c) we can learn that $\Phi_{k\times r}M_C = \tilde{\Phi}$. Therefore, we have $\Phi_{k\times r}\Delta_C = 0$ and prove Theorem 4.1. □

From Theorem 4.1 we can also derive a very useful corollary, as shown below.

**Corollary 4.2.** *If $\Delta_{C_0}$, $\Delta_{C_1}$, ..., $\Delta_{C_n}$ are valid perturbation vectors that follow the constraints, then we have $\Delta_{C'} = \sum_{i=0}^{n} a_i \cdot \Delta_{C_i}$ is also a valid perturbation for the constraint $\Phi_{k\times r}$.*

*Proof.* We have $\Phi_{k\times r}\Delta_{C'} = \Phi_{k\times r}\sum_{i=0}^{n} a_i \cdot \Delta_{C_i} = \sum_{i=0}^{n} a_i \cdot \Phi_{k\times r}\Delta_{C_i}$. Since $\Delta_{C_i}$ is a valid perturbation vector and $\Phi\Delta_{C_i} = 0$, we have $\Phi_{k\times r}\Delta_{C'} = 0$ and prove Corollary 4.2. □

Theorem 4.1 indicates that the perturbation vector to be added to the original measurements must be a solution of the homogeneous linear equations $\Phi_{k\times r}X = 0$. However, is this condition always met? We present Theorem 4.3 to answer this question,.

**Theorem 4.3.** *In practical scenarios, the attacker can always find a valid solution (perturbation) that meets the linear equality constraints imposed by the physical systems.*

*Proof.* Due to the intrinsic property of the targeted system, equation (3c) is naturally met, which indicates that there is always a solution for the nonhomogeneous linear equations $\Phi_{k\times r}X = \tilde{\Phi}$. Accordingly, we have $Rank(\Phi_{k\times r}) \leq r$. Moreover, if $Rank(\Phi_{k\times r}) = r$, there will be one unique solution for equation (4.3c), which means the measurements of compromised sensors are constant. The constant measurements are contradictory to the purpose of deploying CPSs. In practical scenarios, $M$ is changing over time, so that $Rank(\Phi_{k\times r}) < r$ and the homogeneous linear equation $\Phi_{k\times r}X = 0$ will have infinite solutions. Therefore, the attacker can always build a valid adversarial example that meets the constraints.

□

We utilize a simplified example to illustrate how the constraints will affect the generation of perturbations, as shown in Figure 4.5. According to 4.5, measurement $M$ should move a small step (perturbation) to the gradient direction (direction 1 in Figure 4.5) to increase the

loss most rapidly. However, as shown by the contour lines in Figure 4.5, the measurement $M$ is always forced to be on the straight line $y = 2 - 2x$, which is the projection of the intersection of the two surfaces. Accordingly, instead of following the raw gradient, $M$ should move forward to direction 2 to increase the loss. Therefore, although at a relatively slow rate, it is still possible for the attacker to increase the loss under the constraints.

### 4.4.3 Adversarial Example Generation under Linear Equality Constraint

The common method of solving optimization problems using gradient descent under constraints is projected gradient descent (PGD). However, since neural networks are generally not considered as convex functions [21], PGD cannot be used to generate adversarial examples directly. We propose the design of a simple but effective search algorithm to generate the adversarial examples under physical linear equality constraints.

**Algorithm 4:** Best-Effort Search (Linear Equality)

**Input:** $\Delta$, $f_\theta$, $C$, $M$, $step$, $size$, $\Phi$, $Y$
**Output:** $v$
**function** genEqPer($\Delta, f_\theta, C, M, step, size, \Phi, Y$)
    initialize $v = \Delta$
    initialize $stepNum = 0$
    **while** $stepNum \leq step - 1$ **do**
        **if** $f'_{\theta'}(M + v)$ *doesn't equals* $Y$ **then**
            **return** $v$
        **end**
        $r =$ **eqOneStep**$(f_\theta, C, M + v, size, \Phi, Y)$
        update $v = v + r$
        $stepNum = stepNum + 1$
    **end**
    **return** $v$
**end**

As discussed in subsection 4.4.2, the perturbation $\Delta_C$ needs to be a solution of $\Phi_{k\times r}X = 0$. We use $n = Rank(\Phi_{k\times r})$ to denote the rank of the matrix $\Phi_{k\times r}$, where $0 < n < r$. It is obvious that the solution set of homogeneous linear equation $\Phi_{k\times r}X = 0$ will have $r - n$ basic solution vectors. We use $I = [i_0, i_1, ..., i_{r-n-1}]^T$ to denote the index of independent

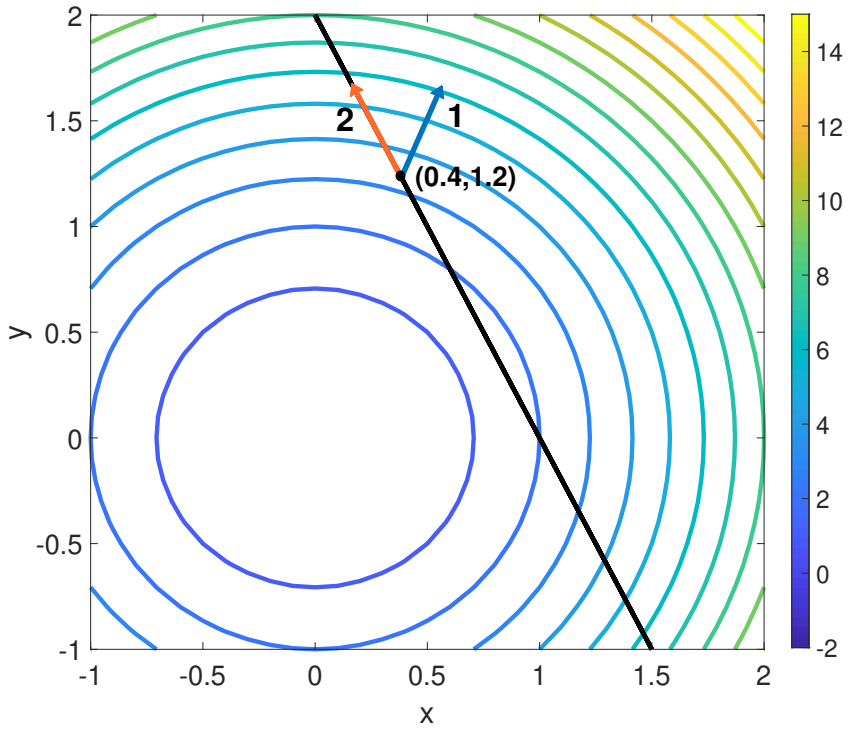


**Figure 4.5:** Linear equality constraint illustration.

variables in the solution set, $D = [d_0, d_1, ..., d_{n-1}]^T$ to denote the index of corresponding dependent variables, and $B_{n\times(r-n)}$ to denote the linear dependency matrix of $X_I$ and $X_D$. Clearly, we have $X_{D_{n\times 1}} = B_{n\times(r-n)}X_{I_{(r-n)\times 1}}$. For convenience, we will use $[I, D, B] =$ **dependency**$(\Phi_{k\times r})$ to describe the process of getting $I$, $D$, $B$ from matrix $\Phi_{k\times r}$.

As shown in Algorithm 4, the function **genEqPer** takes $\Delta$ as an input and outputs a valid perturbation $v$ for $M$. Algorithm 4 keeps executing **eqOneStep** for multiple times defined by *step* to generate a valid $v$ increasingly. Function **eqOneStep** performs a single-step attack for the input vector and returns a one-step perturbation $r$ that matches the constraints defined by $\Phi$, which is shown in Algorithm 5. Due to Corollary 4.2, $\Delta$ and $v$ will also follow the constraints. To decrease the iteration time, similar to [75], the algorithm will return the crafted adversarial examples immediately as long as $f'_{\theta'}$ misclassifies the input measurement vector $M + v$, as shown by Line 7 in Algorithm 3.

**Algorithm 5:** One Step Attack Constraint $\Delta_C$

```
1  Input: f_θ, C, M, size, Φ, Y
2  Output: r
3  function eqOneStep(f_θ, C, M, size, Φ_{k×r}, Y)
4      calculate gradient vector G = ∇_M L(f_θ(M), Y)
5      set all elements of G_U in G to zero
6      define G' = G_C
7      obtain tuple [I, D, B] = dependency(Φ_{k×r})
8      update G'_D = B G'_I in G'
9      ε = size/max(abs(G'))
10     return r = εG
11 end
```

The philosophy of function **eqOneStep** in algorithm 5 is very straightforward. From the constraint Matrix $\Phi$, we can get the independent variables $I$, dependent variables $D$ and the dependency matrix $B$ between them. We will simply keep the gradient values of $I$ and use them to compute the corresponding values of $D$ (Line 7) so that the final output perturbation $r$ will follow $\Phi$. The constant factor *size* defines the largest modification the attacker can make to a specific measurement to control the search speed.

### 4.4.4 Adversarial Example Generation under Linear Inequality Constraint

**Algorithm 6:** Non-Constraint Perturbation.

**Input:** $f'_{\theta'}$, $U$, $M$, $size$, $Y$
**Output:** $r$
**function** freeStep($f'_{\theta'}, U, M, size, Y$)
    calculate gradient vector $G = \nabla_M L(f'_{\theta'}(M), Y)$
    set elements in $G_U$ to zero
    $\epsilon = size/\mathbf{max}(\mathbf{abs}(G))$
    **return** $r = \epsilon G$
**end**

Linear inequality constraints are very in real-world CPS applications, like the water flow constraints in Figure 4.3. Due to measurement noise, real-world systems usually tolerate distinctions between measurements and expectation values as long as the distinctions are smaller than predefined thresholds, which also brings inequality constraints to data. Meanwhile, a linear equality constraint can be represented by two linear inequality constraints. As shown in equation (4.4), linear inequality constraints define the valid measurement subspace whose boundary hyper-planes are defined by equation (4.2). In general, the search process under linear inequality constraints can be categorized into two situations. The first situation is when a point (measurement vector) is in the subspace and meets all constraints, while the second situation happens when the point reaches boundaries.

Due to the property of physical systems, the original point $M$ will naturally meet all the constraints. As shown in Algorithm 7, to increase the loss, the original point will first try to move a step following the gradient direction through the function **freeStep** defined in Algorithm 6. Algorithm 6 is very similar to the FGM algorithm [89] but no perturbation is added to $M_U$, namely $r_U = 0$, which is similar to the saliency map function used in [84]. After that, the new point $M'$ is checked with equation (4.4) to find if all inequality constraints are met. If all constraints were met, the moved step was valid and we can update $M = M'$. If $M'$ violates some constraints in $\Phi$, we will take all the violated constraints and make a real-time constraint matrix $\Phi_V$, where $V$ is the index vector of violated constraints. We now convert the inequality constraint problem to the equality constraint problem with

**Algorithm 7:** Best-Effort Search (Linear Inequality)

**Input:** $\Delta$, $f'_{\theta'}$, $C$, $U$, $M$, $step$, $size$, $\Phi$, $\tilde{\Phi}$, $Y$
**Output:** $v$
**function** genIqPer($\Delta, f'_{\theta'}, C, U, M, step, size, \Phi, \tilde{\Phi}, Y$)
    initialize $pioneer = \Delta$, $valid = pioneer$
    initialize $stepNum = 0$
    initialize $V$ as empty // violated constrain index
    **while** $stepNum \leq step - 1$ **do**
        **if** $f'_{\theta'}(M + valid)$ *doesn't equals* $Y$ **then**
            **break**
        **end**
        $chkRst =$ **chkIq**$(\Phi, \tilde{\Phi}, M + pioneer, C)$
        **if** $chkRst$ *is empty* **then**
            $valid = pioneer$
            $r =$ **freeStep**$(f'_{\theta'}, U, M + valid, size, Y)$
            $pioneer = valid + r$
            **reset** $V$ to empty
        **else**
            extend $V$ with $chkRst$
            **define** $\Phi' = \Phi_V$ // real-time constraints
            $r =$ **eqOneStep**$(f'_{\theta'}, C, M + valid, size, \Phi', Y)$
            $pioneer = valid + r$
        **end**
        $stepNum = stepNum + 1$
    **end**
    **return** $v = valid$
**end**

the new constraint matrix $\Phi_V$ and the original point $M$. $M$ will then try to take a step using the **eqOneStep** function described in Algorithm 5 with the new constraint matrix $\Phi_V$. Again, we check whether the new reached point meets all the constraints. If there are still violated constraints, we extend $V$ with the new violated constraints. The search process repeats until reaching a valid $M'$ that meets all the constraints. For simplicity, we will use $chkRst = \textbf{chkIq}(\Phi, \tilde{\Phi}, M', C)$ to denote the checking process of a single search in one step movement, where $chkRst$ is the index vector of the violated constraints in the search.

Similar to Figure 4.5, a simple example is shown in Figure 4.6. To increase the loss, the initial point $a$ will take a small step following the gradient direction and reach point $b$. Since $b$ meets the constraints, it is a valid point. After that, $b$ will move a step following the gradient direction and reach point $c'$. However, point $c'$ violates the constraint $\beta$ and the movement is not valid. As we have point $b$ is valid, we construct a linear equality constraint problem with constraint $\alpha$ which is parallel to $\beta$. With constraint $\alpha$, point $b$ will move a step to point $c$ which is also a valid point. Point $c$ then repeats the search process and increases the loss gradually. The real-time equality constraint is only used once. When a new valid point is reached, it empties the previous equality constraints and tries the gradient direction first.

## 4.5 Experimental Evaluation

In this section, we evaluate our ConAML frameworks in CPS control domain with three different CPS applications, including incident detection in transportation system, false data injection attack in power system state estimation, and anomaly detection in water treatment systems. We analyze and examine the practical requirements for launching adversarial attacks in the three CPS applications, and summarize the corresponding constraints in Table 4.2.

From Table 4.2, we can learn that the attacker needs to overcome all constraints for power grids and water systems CPSs in our study case, while the knowledge constraint is released for the transportation study case.

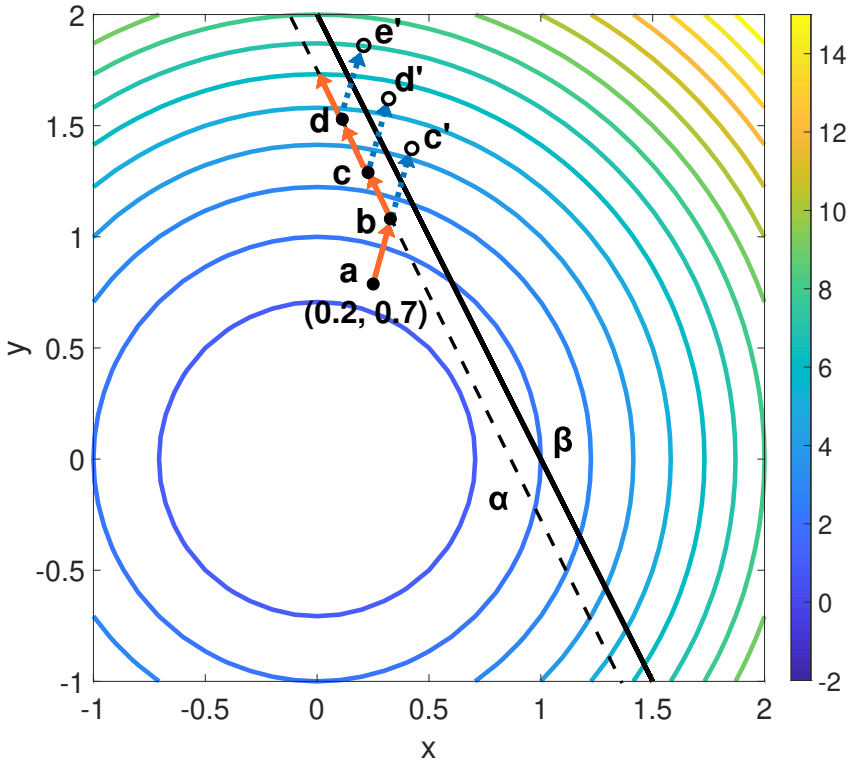


**Figure 4.6:** Best-Effort Search (linear inequality).

**Table 4.2:** Study Case Constraints

| Constraints | Transportation | Power Grids | Water Systems |
|---|---|---|---|
| Model Constraint | ★ | ★ | ★ |
| Sensor Constraint | ★ | ★ | ★ |
| Knowledge Constraint | | ★ | ★ |
| Physical Constraint | ★ | ★ | ★ |
| Time Constraint | ★ | ★ | ★ |

We will present the detailed attack scenarios for each study case in the subsections. The deep learning models are implemented using Tensorflow and the Keras library and are trained on a Windows 10 machine with an Intel i7 CPU.

### 4.5.1 Case Study: Incident detection in transportation systems

#### Background: Deep learning-based incident detection

Traffic incidents can be a great threat to people's lives and property, and timely incident detection is very important for life-saving. On the other hand, with the development of the Intelligent Transportation System (ITS), different sensing techniques are employed in highways and provides massive heterogeneous data that contains the real-time traffic information, such as average speed, total traffic flow, and average occupancy. With big data techniques, the high granularity traffic data can be taken advantage of for many important applications in ITS, such as traffic prediction [117, 64, 100] and incident detection.

In recent years, deep neural networks have been widely studied to be the key techniques for incident detection [40, 62, 115, 87, 86, 120] . For example, an LSTM model can learn the time-series pattern of the traffic information changing when an incident happens. With fine tunned parameters, the DNNs achieve state-of-the-art performance in transportation system incident detection.

#### Adversarial attacks for DNN-based incident detection

In general, the incident detection DNNs can be considered as a binary classifier, with the input features contains the traffic information (speed). The DNN will predict if there is an incident in the given timeslot based on the real-time speed data. The main metrics to evaluate the performance of a trained incident detection DNN is are detection rate and false alarm.

The state-of-the-art models in the literature can achieve an around 90% detection rate and below around 10% false alarm rate. In this study case, we consider the attack scenario that a malicious attacker aims to disable the availability of the DNN-based incident detection in a transportation system. The adversarial attacker would launch a false-positive attack

that aims to deceive the incident detection DNN to predict the wrong result (incident) based on the normal traffic data.

As demonstrated in Table 4.2, the adversarial attacker of incident detection needs to overcome model constraint, sensor constraint, physical constraint, and time constraint. We assume the attacker can compromise a portion of speed sensors in a highway and can freely modify the measurements of the compromised sensors. Meanwhile, due to the continuous property of the highway, the difference between the speed measurements of adjacent sensors should be small, namely the **Physical Constraint** is $\|S_{i,t} - S_{i-1,t}\| \leq \epsilon$, where $\epsilon$ is a constant defined by the traffic condition. However, many transportation systems make the real-time speed measurements public available, which enables the attacker to know the real-time measurements of uncompromised speed sensors. Therefore, there is no knowledge constraint for the adversarial attacker who targets transportation system CPSs.

### Simulation Evaluation

**Dataset:** Different datasets were used in previous literature. In this study case, we study the traffic data of US I-880 N highway. On the one hand, it was used in previous research [30, 40]. On the other hand, the California Department of Transportation provides a convenient public portal for traffic data collection, and the incident data and traffic data of I-880 N highway are publicly available on the Caltrans Performance Measurement system (PeMS) [82].

We collected all the I-880 N incident data of the year 2017, the incident types in the data include traffic collision, hit and run injury, car fire, traffic break, animal hazard, and construction. We collect 7,111 incident data records in total. We then collected the related traffic speed data from 98 sensor stations in I-880 N. The sensors report the average speed of the monitoring every five minutes. For incident that happened in slot $n$, we collected the corresponding speed data from time slot $n-2$ to time slot $n+2$, and the regulated speed data structure is demonstrated by Fig. 4.7. We label all incident traffic data records as 0. We then randomly sample 7,111 normal traffic data records and label them as 1.

**Incident detection DNN:** LSTM DNNs are widely employed for incident detection in previous literature since they have intrinsic advantages in learning the time-series data [115, 40]. In this case study, we train two LSTM DNNs for the defender and the attacker with

the regulated DNN described above for traffic data classification, and the structures of two DNNs are demonstrated in Tabel 4.3. We randomly split the dataset into the training part (85%) and the testing part (15%). Both models are trained with a 0.0001 learning rate, 512 batch size, a mean squared error loss function, and a Stochastic Gradient Descent (SGD) optimizer. Finally, the defender's DNN achieves an 87.4% detection rate with 10.1% false alarm rate while the attacker DNN's detection rate is 86.5% and the false alarm rate is 9.7%.

**Adversarial Attacks:** We examine the traffic speed data of I-880 N highway, and set the $\epsilon = 8.5$ for the physical constraint. The I-880 N highway has 98 speed sensors in the DNN systems in total. In our simulations, we assume there are 5, 10, 15, 20 sensors being compromised by the adversarial attacker respectively, and the sensors are randomly selected. We launch the adversarial attack to the defender's detection DNN with our ConAML algorithm, and the result is demonstrated in Table 4.4.

From Table 4.4, we can learn that our adversarial attack can significantly increase the false positive rate of the detection DNN with small modifications to the speed measurements. In general, with more sensors being compromised, the attacker can have a better attack performance.

### 4.5.2 Case Study: False Data Injection Attack Detection in Power System State Estimation

#### Background: State Estimation and FDIA

Power grids are critical infrastructures that connect power generation to end customers through transmission and distribution grids. In recent decades, the rapid development of technologies in sensors, communication, and computing enables various applications in the power grid. However, as the power system becomes more complex and dependent on the information and communications technology, the threat of cyber-attacks also increases, and the cyber-power system becomes more vulnerable [98, 101]. The cyberattack to the Ukraine power grid in 2015 is a well-known example [63].

$$\begin{bmatrix} S_{0,t-2} & S_{0,t-1} & S_{0,t} & S_{0,t+1} & S_{0,t+2} \\ S_{1,t-2} & S_{1,t-1} & S_{1,t} & S_{1,t+1} & S_{1,t+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ S_{97,t-2} & S_{97,t-1} & S_{97,t} & S_{97,t+1} & S_{97,t+2} \end{bmatrix}$$

**Figure 4.7:** Speed Data Structure

**Table 4.3:** Incident Detection LSTM

| **Layer** | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **Defender** | 16 LSTM | 16 LSTM | 0.25 Dropout | 2 Softmax |
| **Attacker** | 24 LSTM | 0.25 Dropout | 2 Softmax | - |

**Table 4.4:** Incident Detection Evaluation Result

| **Case** | False Positive Rate | Ave $L_1$-Norm (mile/h) |
|---|---|---|
| **5** | 82.8% | 5.9 |
| **10** | 71.4% | 6.3 |
| **15** | 83.7% | 7.1 |
| **20** | 87.7% | 2.8 |

State estimation is a backbone of various crucial applications in power system control that has been enabled by large scale sensing and communication technologies, such as SCADA. Generally speaking, the state estimation is used to estimate the state of each bus, such as voltage angles and magnitudes, in the power grid through analyzing other measurements.

We denote the vector of state variables as $\mathbf{x} = [x_1, x_2, ..., x_n]^T$, and the meters' measurements vector as $\mathbf{z} = [z_1, z_2, ..., z_m]^T$, where $x_i \in R$ and $z_j \in R$. The general state estimation process can then be represented as follow:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e} \tag{4.6}$$

where $\mathbf{e}$ is the measurement error vectors and $\mathbf{h}$ is a function of $\mathbf{x}$. In practice, a simplified DC power flow state estimation can be used to decrease the process time cost. A DC model can then be represented as equation (4.7).

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e} \tag{4.7}$$

The matrix $\mathbf{H}_{m \times n}$ is determined by the configurations, topology and physical parameters of the power system.

In general, a weighted least squares estimation (WLS) approach is used to solve equation (4.7). The estimated state vector $\hat{\mathbf{x}}$ can then be computed through equation (4.8):

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{z} \tag{4.8}$$

where matrix $\mathbf{W}$ is the covariance matrix of the meters' statistical measurement errors.

The measurements $\mathbf{z}$ may contain bad measurements due to possible meter errors or cyber attacks. Therefore, state estimation usually integrates with a linear residual-based detection approach to remove faulty measurements according to the difference between $\mathbf{z}$ and $\mathbf{H}\hat{\mathbf{x}}$. If the $L_2$-norm of $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|$ is larger than a threshold $\tau$ that is selected according to a false alarm rate, the measurement $\mathbf{z}$ will be considered as polluted and be removed.

The residual-based detection involves non-linear computation ($L_2$-Norm), however, research has shown that a false measurement vector follows linear equality constraints can be used to pollute the normal measurements without being detected. In 2009, Liu *et al.*

proposed the false data injection attack (FDIA) that can bypass the residual-based detection scheme and finally pollute the result of state estimation [66]. In particular, if the attacker knows $\mathbf{H}$, she/he could construct a faulty vector $\mathbf{a}$ that meets the linear constraint $\mathbf{Ba} = \mathbf{0}$, where $\mathbf{B} = \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T - \mathbf{I}$, and the crafted faulty measurements $\mathbf{z}+\mathbf{a}$ will not be detected by the system, as demonstrated below.

The FDIA enables an attacker to generate a false measurement vector $\mathbf{a} = [a_1, a_2, ..., a_m]^T$ to be added to legitimate measurement $\mathbf{z}$, so that the polluted measurements will be $\mathbf{z}_a = \mathbf{z} + \mathbf{a}$. The original FDIA proposed in [66] shows that if the attacker knows the matrix $\mathbf{H}$, she/he can construct $\mathbf{a} = \mathbf{Hc}$ ($\mathbf{c}$ represents the estimation error) that can bypass the fault detection in state estimation, as shown by equation (4.9), where $\hat{\mathbf{x}}_{bad}$ and $\hat{\mathbf{x}}$ denote the estimated $\mathbf{x}$ using $\mathbf{z}_a$ and $\mathbf{z}$ respectively.

$$\begin{aligned} \|\mathbf{z}_a - \mathbf{H}\hat{\mathbf{x}}_{bad}\| &= \|\mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c})\| && (4.9a) \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{a} - \mathbf{Hc})\| && (4.9b) \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| \leq \tau && (4.9c) \end{aligned}$$

Meanwhile, equation (4.10) from [66] provides an efficient way to generate a valid vector $\mathbf{a}$, where $\mathbf{P} = \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ and matrix $\mathbf{B} = \mathbf{P} - \mathbf{I}$.

$$\begin{aligned} \mathbf{a} = \mathbf{Hc} &\Leftrightarrow \mathbf{Pa} = \mathbf{PHc} \Leftrightarrow \mathbf{Pa} = \mathbf{Hc} \Leftrightarrow \mathbf{Pa} = \mathbf{a} && (4.10a) \\ &\Leftrightarrow \mathbf{Pa} - \mathbf{a} = \mathbf{0} \Leftrightarrow (\mathbf{P} - \mathbf{I})\mathbf{a} = \mathbf{0} && (4.10b) \\ &\Leftrightarrow \mathbf{Ba} = \mathbf{0} && (4.10c) \end{aligned}$$

Equation (4.10c) indicates that $\mathbf{a}$ is a solution of the homogeneous equation $\mathbf{BX} = \mathbf{0}$. If an attacker compromised $k$ measurements in $\mathbf{z}$, and there will be $k$ non-zero elements in $\mathbf{a}$. Equation (4.10c) will then become:

$$\mathbf{B'a'} = \mathbf{0} \tag{4.11}$$

where $\mathbf{B'}_{m\times k}$ and $\mathbf{a'}_{m\times k}$ are corresponding columns and rows sampled from $\mathbf{B}$ and $\mathbf{a}$ respectively according to the $k$ compromised measurements. Liu *et al.* proved that as long as $k > m - n$, non-zero $\mathbf{a}$ always exists.

As FDIA presented a serious threat to the power grid security, many detection and mitigation schemes to defend FDIA are proposed, including strategical measurement protection [12] and PMU-based protection [112]. In recent years, detection schemes based on ML, especially neural networks, have been proposed and become popular [83, 111, 44, 47, 10, 81, 105] in the literature. The ML-based detection does not require extra hardware equipment and achieve the state-of-the-art detection performance. However, in this section, we will demonstrate that ML approaches are vulnerable to ConAML. The ML models in previous research are trained to distinguish normal measurement $\mathbf{z}$ and poisoned measurement $\mathbf{z} + \mathbf{a}$. Our ConAML algorithms allow the attacker to generate an adversarial perturbation $\mathbf{v}$ that meets the constraint $\mathbf{Bv} = \mathbf{0}$ for his/her original false measurement $\mathbf{z} + \mathbf{a}$ and obtain a new false measurement vector $\mathbf{z}_{adv} = \mathbf{z} + \mathbf{a} + \mathbf{v}$ that will be classified as normal measurements by the ML-based FDIA detection models. The matrix $\mathbf{B}$ then acts as the constraint matrix $\Phi$ defined in equation (4.3). Meanwhile, $\mathbf{z}_{adv}$ can naturally bypass the traditional residual-based detection approach since the total injected false vector $\mathbf{a} + \mathbf{v}$ meets the constraint $\mathbf{B}(\mathbf{a} + \mathbf{v}) = \mathbf{Ba} + \mathbf{Bv} = \mathbf{0}$. Our experiment in the next subsection will show that our ConAML algorithms can significantly decrease the detection performance of the ML-based detection schemes.

### Experiment Design and Evaluation

We select the IEEE standard 10-machine 39-bus system as the power grid system as it is one of the benchmark systems in related research. The structure of the IEEE 39-bus system is shown in Figure 4.8. The features used for ML model training are the power flow (Ampere) measurements of each branches. The system has 46 branches so that there there will be 46 features for the ML models.

In our experiment, the goal of the attacker is to implement a false negative attack that makes the polluted measurements $\mathbf{z}_{adv}$ pass the detection of the ML models, namely to fool the models to misclassify the false measurements as normal.

We utilize the MATPOWER [121] library to derive the $\mathbf{H}$ matrix of the system and simulate the power flow measurement data. We also implement the FDIA using MATLAB to generate false measurements. Both the power flow measurements and false measurements follow Gaussian distributions. We make two datasets for the defender and the attacker respectively. For each dataset, there are around 25,000 records with half records are polluted with FDIA. We label the normal measurements as 0 and false measurements as 1 and use one-hot encoding for the labels.

We investigate the scenarios that there are 10, 13, and 15 measurements being compromised by the attacker, with the randomly generated compromised index vector $C$ and corresponding constraint matrix $\Phi$ ($\mathbf{B}_C$ in (4.11)). We generate 1,000 false measurement vectors in each test datasets.

After that, we train two deep learning models based on the training datasets accordingly, with 75% records in the dataset used for training and 25% for testing. We use simple fully connected neural networks as the ML models, as shown in Table 4.5. Both the models are trained with a 0.0001 leaning rate, 512 batch size, and a mean squared error loss function. The training process is around one minutes for each model.

Table 4.6 summarizes the detection performance of $f_\theta$ under adversarial attacks generated by our ConAML algorithms. From the table, we can learn that the ConAML attacks can effectively decrease the detection accuracy of the ML models used for FDIA detection and inject considerable bad data to the power systems. In all three study cases, the detection accuracy of the defender's model decreased to below 30% under the adversarial attacks. Meanwhile, we can observe that the $L_2$-Norm are very large, especially for the '15' study case.

As shown in Figure 4.9, by comparing the evaluation results of different cases, we can learn that compromising more sensors cannot guarantee better performances in attack detection. This is due to the different physical constraints imposed by the system. However, with more compromised sensors, the attacker can usually obtain a larger size of the injected bad data.

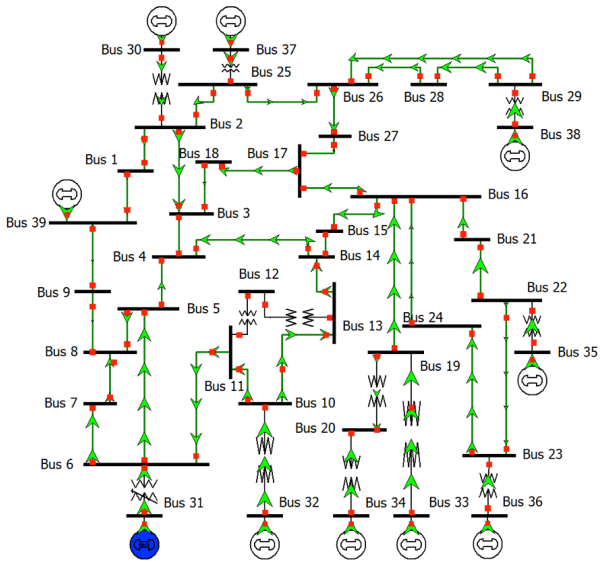


**Figure 4.8:** IEEE 39-Bus System [8] [29].

**Table 4.5:** Model Structure - FDIA

| Layer | $f$ | $f'$ |
|---|---|---|
| 0 | 46 Input | 46 Input |
| 1 | 32 Dense ReLU | 30 Dense ReLU |
| 2 | 48 Dense ReLU | 40 Dense ReLU |
| 3 | 56 Dense ReLU | 30 Dense ReLU |
| 4 | 48 Dense ReLU | Dropout 0.25 |
| 5 | 32 Dense ReLU | 20 Dense ReLU |
| 6 | Dropout 0.25 | Dropout 0.25 |
| 7 | 16 Dense ReLU | 2 Dense Softmax |
| 8 | Dropout 0.25 | - |
| 9 | 2 Dense Softmax | - |

**Table 4.6:** Evaluation Result Summary

| **Attack** | **Case** | **Accu** | **$L_2$-Norm** | **Time (ms)** |
|---|---|---|---|---|
| black-box | 10 | 14.4% | 1843.2 | 131.9 |
| | 13 | 4.3% | 4786.72 | 209.6 |
| | 15 | 28.1% | 9079.02 | 163.3 |

In our experiments, the time cost is relatively higher due to the universal adversarial measurements algorithm, as shown in Figure 4.10. However, the time cost is still efficient for many CPS applications in practice. For example, the sampling period of the traditional SCADA system used in the electrical power system is 2 to 4 seconds. In practical scenarios, the time cost of adversarial example generation depends on the computational resource of the attacker. With the possible optimization and upgrade in software and hardware, the time cost can be further reduced.

### 4.5.3 Case Study: Water Treatment System

#### Background: SWaT Dataset

We study the linear inequality physical constraints based on the Secure Water Treatment (SWaT) proposed in [34]. SWaT is a scaled-down system but with fully operational water treatment functions. The testbed has six main processes and consists of cyber control (PLCs) and physical components of the water treatment facility. The SWaT dataset, generated by the SWaT testbed, is a public dataset to investigate the cyber attacks on CPSs. The raw dataset has 946,722 samples with each sample comprised of 51 attributes, including the measurements of 25 sensors and the states of 26 actuators. Each sample in the dataset was labeled with normal or attack. The detailed description of the SWaT dataset can be found in [34] and [56].

The SWaT dataset is an important resource to study anomaly detection in CPSs. In 2017, Inoue *et al.* used unsupervised machine learning, including Long Short-Term Memory (LSTM) and SVM, to perform anomaly detection based on the SWaT dataset [45]. By comparison, Kravchik *et al.* employed Convolutional Neural Networks (CNN) and achieved a better false positive rate [53]. In 2019, [28] proposed a data-driven framework to derive invariant rules for anomaly detection for CPS and utilized SWaT to evaluate their approach. Other research related to the SWaT dataset can be found in [18, 27, 5].

As shown in Table 4.7, the SWaT dataset includes the measurements from five kinds of analog components (25 sensors in total) whose measurements are used as the input features in previous anomaly detection ML models. Our experiments aims to demonstrate that the

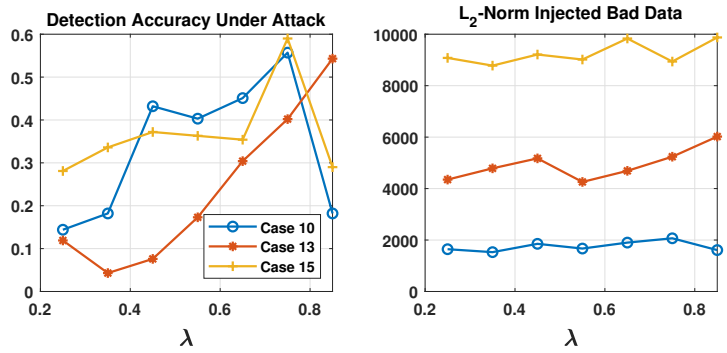


**Figure 4.9:** Performance of black-box attacks according to $\lambda$ with $step = 40$, $size = 20$.

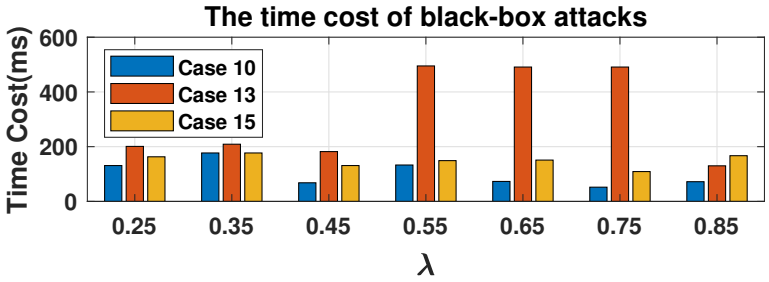


**Figure 4.10:** Time cost of black-box attacks according to $\lambda$ with $step = 40$, $size = 20$.

**Table 4.7:** SWaT Analog Components

| Symbol | Description | Unit |
|---|---|---|
| **LIT** | Level Indication Transmitter | $mm$ |
| **FIT** | Flow Indication Transmitter | $m^3/hr$ |
| **AIT** | Analyzer Indication Transmitter | $uS/cm$ |
| **PIT** | Pressure Indication Transmitter | $kPa$ |
| **DPIT** | Differential Pressure Ind Transmitter | $kPa$ |

ML models used for anomaly detection are vulnerable to adversarial attacks. However, due to the physical properties of the SWaT testbed, the sensor's measurements are not independent but with linear inequality constraints.

In our experiment, we consider the scenario that the adversarial attacker compromises the **FIT** components to inject bad adversarial water flow measurements while bypassing the ML-based anomaly detection system. We then examined the SWaT testbed structure and find out that there are apparent linear inequality constraints among the **FIT** measurements. We checked the SWaT dataset and observed that all the normal examples in the dataset meet the constraints. We also contacted the managers of the SWaT testbed and verified our find. The linear inequality constraints of the seven **FIT** measurements in the dataset are defined by the structure of the water pipelines and the placement of the sensors, as shown in equation 4.12, where $\epsilon_1$ and $\epsilon_2$ are two constants of the system's noise tolerance. We utilized the double value of the maximum difference of the corresponding measurements in the SWaT dataset to estimate $\epsilon_1$ and $\epsilon_2$, and we had $\epsilon_1 = 0.0403$ and $\epsilon_2 = 0.153$. Therefore, the adversarial measurements should also follows the same linear inequality constraints to avoid being noticed by the system operator.

$$\mathbf{FIT301} \leq \mathbf{FIT201} \tag{4.12a}$$

$$\|\mathbf{FIT401} - \mathbf{FIT501}\| \leq \epsilon_1 \tag{4.12b}$$

$$\|(\mathbf{FIT502} + \mathbf{FIT503}) - (\mathbf{FIT501} + \mathbf{FIT504})\| \leq \epsilon_2 \tag{4.12c}$$

Based on (4.4), we can represent (4.12) as follow. And $M_C$ is the vector of measurements of **FIT201**, **FIT301**, **FIT401**, **FIT501**, **FIT502**, **FIT503** and **FIT504** accordingly.

### Experimental Design and Evaluation

Similar to the power system study case, we generate two training datasets for the defender's model $f_\theta$ and the attacker's model $f'_{\theta'}$ respectively by poisoning the normal measurements

with Gaussian noise. The ML models is trained to distinguish the normal measurement data and the poisoned measurements (anomaly).

$$\Phi_{5\times 7} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 & -1 & 1 \end{bmatrix} \tilde{\Phi} = \begin{bmatrix} 0 \\ 0.0403 \\ 0.0403 \\ 0.153 \\ 0.153 \end{bmatrix}$$

In the Swat dataset, we extracted the normal records which were sampled when the whole system was working steadily. We also removed all the actuators' features. Here, we denote the extracted records as $D_e$. After that, we randomly picked out three test datasets from $D_e$ as the with each test dataset contains 1000 records. We added Gaussian noise to the compromised measurements of records in all test datasets. We checked the polluted record every time when a noise vector was added to ensure all the records in test datasets meet the linear inequality constraints. Here, we denote the rest records of $D_e$ as $D_{train}$ which contains 120,093 records with each record having 25 features in our implementation. We randomly and equally split $D_{train}$ into $D_{defender}$ and $D_{attacker}$ for the defender and attacker respectively and pollute half records with normally-distributed random noise in $D_{train}$ and $D_{defender}$. The polluted records in $D_{defend}$ and $D_{attacker}$ are labeled with 1 and the rest with 0. We allow the records in $D_{train}$ and $D_{defend}$ with label 1 to violate the constraints since the ML models are also expected to detect the obviously anomalous measurements.

We utilize $D_{defend}$ and $D_{attack}$ to train the ML models $f_\theta$ and $f'_{\theta'}$ for the defender and attacker respectively. Again, 75% records in the both datasets were used for training the 25% records for testing during the training process. Similar to FDIA experiment, we utilize fully connected neural networks and the structures are shown in Table 4.8. Through parameter tuning, model $f_\theta$ and $f'_{\theta'}$ achieves 97.2% and 96.7% accuracy respectively.

After that, we consider the scenarios that there were 2, 5, and 7 **FIT** measurements compromised by the attacker and generate the related test datasets. The goal of the attacker is to generate the adversarial **FIT** measurements with the linear inequality constraints

defined by equation (4.12) so that the poisoned measurements can be classified as 'normal' by the defender's ML model $f_\theta$.

Table 4.9 summarizes the evaluation performances of ConAML attacks. From the table, we can learn that the ConAML framework can still effectively decrease the detection accuracy of the ML models for black-box attacks. Similar to the power system study case, a larger number of compromised sensors cannot produce a better performance in bypassing the detection. The reason for this result is that more compromised sensors will also have more complex constraints between their measurements. Meanwhile, more constraints will increase the computation overhead of the best effort search algorithms since there will be a 'larger' constraint matrix.

Figure 4.11 demonstrated the trend of the detection accuracy and injected bad data size according to $\lambda$. From the figure, we can learn that, with the $\lambda$ increases, the probability of the adversarial examples being detected also increases. This matches the intuition that if an adversarial example can obtain higher successful attack probability with the sampling measurement set, the probability of evading detection will also increase. Meanwhile, a smaller injected data size is expected to make the adversarial examples look more 'normal' to the detection model.

## 4.6 Extension: Non-Linear Constraints

Many other machine learning-based applications in the CPS domains, for instance, load forecasting in power and water systems, traffic forecasting in transportation systems, have nonlinear constraints. The non-linear constraints can be very complex in various CPSs and cannot be covered in one study.

In general, similar to linear constraints, the $k$ nonlinear constraints of the compromised measurements can be represented as equation (4.13), where $\mu_i$ is a nonlinear function of $M_C$.

We now investigate a special case of the nonlinear constraints. If there exists a subset of the compromised measurements, in which each measurement can be represented as an explicit function of the measurements in the complement set, the attacker will also be able

**Table 4.8:** Model Structure - Water Treatment

| Layer | $f$ | $f'$ |
|---|---|---|
| 0 | 25 Input | 25 Input |
| 1 | 20 Dense ReLU | 24 Dense ReLU |
| 2 | 40 Dense ReLU | 32 Dense ReLU |
| 3 | 30 Dense ReLU | 32 Dense ReLU |
| 4 | Dropout 0.25 | 16 Dense ReLU |
| 5 | 20 Dense ReLU | 2 Dense Softmax |
| 6 | Dropout 0.25 | - |
| 7 | 2 Dense Softmax | - |

**Table 4.9:** Evaluation Result Summary

| **Attack** | **Case** | **Accu** | **$L_2$-Norm** | **Time (ms)** |
|---|---|---|---|---|
| black-box | 2 | 1.3% | 0.309 | 17.5 |
| | 5 | 2.3% | 0.340 | 111.7 |
| | 7 | 1.14% | 0.411 | 451.8 |

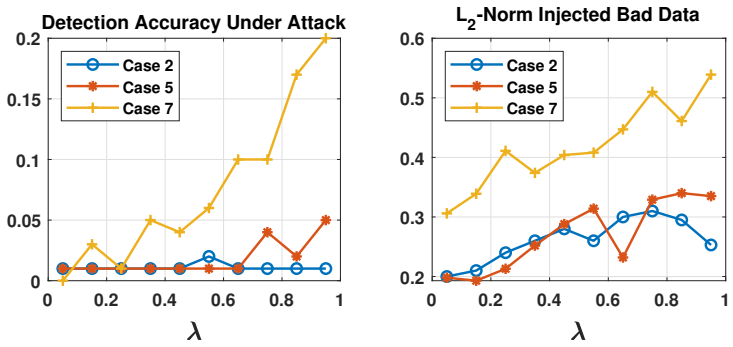


**Figure 4.11:** Performance of black-box attacks according to $\lambda$ with $step = 50$, $size = 0.06$.

to generate the perturbation accordingly. We use $P = [p_0, p_1, ..., p_{n-1}]$ to denote the index vector of the former measurement set, and use $Q = [q_0, q_1, ..., q_{r-n-1}]$ to denote the index vector of the complement set. We can then represent (4.13) as (4.14), where $\Xi = [\xi_0, \xi_1, ..., \xi_{n-1}]$ is a vector of explicit functions.

$$\begin{cases} \mu_0(m_{c_0}, m_{c_1}, ..., m_{c_{r-1}}) = 0 \\ \mu_1(m_{c_0}, m_{c_1}, ..., m_{c_{r-1}}) = 0 \\ ... \\ \mu_{k-1}(m_{c_0}, m_{c_1}, ..., m_{c_{r-1}}) = 0 \end{cases} \tag{4.13}$$

$$\begin{cases} m_{p_0} = \xi_0(m_{q_0}, m_{q_1}, ..., m_{q_{r-n-1}}) \\ m_{p_1} = \xi_1(m_{q_0}, m_{q_1}, ..., m_{q_{r-n-1}}) \\ ... \\ m_{p_{n-1}} = \xi_{n-1}(m_{q_0}, m_{q_1}, ..., m_{q_{r-n-1}}) \end{cases} \tag{4.14}$$

Apparently, the roles of $M_Q$ and $M_P$ in (4.14) are similar to the $M_I$ and $M_D$ in linear constraints correspondingly. Instead of a linear matrix, the function set $\Xi$ represents the dependency between $M_P$ and $M_Q$. The nonlinear constraints make properties such as Theorem 1 infeasible. To meet the constraints, the attacker needs to find the perturbation $\Delta_Q$ first and obtain $M_Q^*$ by adding it to $M_Q$. After that, the attacker can compute $M_P^* = \Xi(M_Q^*)$ .

The above case of nonlinear constraints is special and may not be scalable to various practical applications. Although there are different types of nonlinear systems, they can be generalized using piece-wise linear constraints by setting proper ranges and breakpoints. We leave this as an open problem for future work.

# Chapter 5

# Adversarial Defense in CPS: Random Padding Framework

In this chapter, we study the defense mechanisms for adversarial attacks in cyber-physical systems. We review and study several state-of-the-art defense mechanisms proposed in the computer vision domain, and analyze and evaluate their performance for CPS applications. Meanwhile, we demonstrate that some state-of-the-art adversarial defense methods, such as adversarial detection and input reconstruction, have intrinsic constraints for control domain adversarial attacks. To solve this, we propose a random input padding framework. Simulation evaluation shows that our framework can significantly decrease the effectiveness of adversarial examples in both customer domain (energy theft detection) and control domain (FDIA detection) adversarial attacks.

## 5.1 Defense Requirements

White-box adversarial attacks allow the attacker to have access to the target DNN, which is a common setting in previous literature and has been extensively studied since it helps researchers to learn the weakness of DNNs more directly [108]. Robust against white-box adversarial attacks is the desired property that the DNNs should maintain [104], especially for critical infrastructure. In particular, the cyberattacks targeting critical CPSs are usually nationwide and the attacker owns considerable resources, like the well-known Ukraine power

grid attack in 2015 [63]. In this paper, we expect the defense mechanisms in DNN-based control domain CPS application to be resilient to white-box adversarial attacks.

## 5.2 State-of-the-art Adversarial Defense Mechanisms

In this section, we review several state-of-the-art adversarial defense mechanisms, including adversarial training, adversarial detection, and input reconstruction.

### Adversarial Training

Adversarial training is one of the common methods to mitigate an adversarial attack [54, 91, 104]. The basic principle of adversarial training is to generate and include adversarial examples in each data batch during the training stages. As the DNN is trained to recognize adversarial examples, it becomes more robust.

### Adversarial Detection

Adversarial detection aims to recognize adversarial examples at the DNN inference stage [69, 73, 109]. In particular, a auxiliary binary classification DNN $F_{adv}$ is trained with normal records and corresponding adversarial examples [73] to detect if an input is an adversarial example. The adversarial detection DNN $F_{adv}$ will be employed first to recognize the input records, and only the normal records will be fed into the original functional DNN.

### Input Reconstruction

The input reconstruction mechanisms aim to recover the normal input records from possible adversarial examples [38][72]. Typically, an autoencoder is used to reconstruct the model inputs. Since the autoencoder is trained only with normal data records, it can learn the overall distribution of the normal data. When an adversarial example is received, the autoencoder can push the adversarial example to the manifolds of its legitimate records [72]. Meanwhile, the divergence between the autoencoder's input and output can also be used as a metric for adversarial example detection.

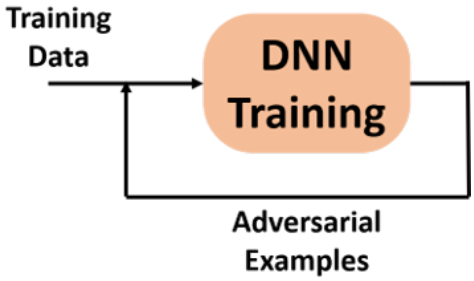


**Figure 5.1:** Adversarial Training

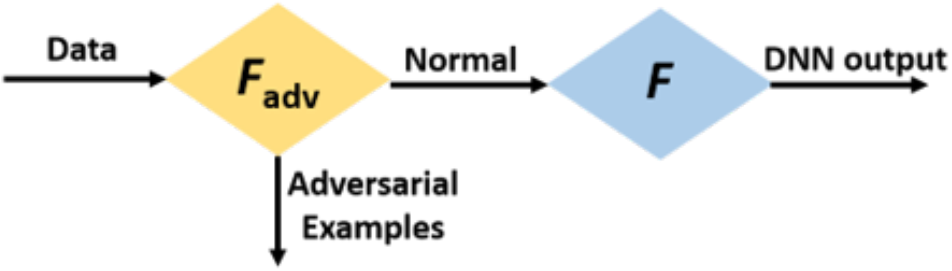


**Figure 5.2:** Adversarial Detection

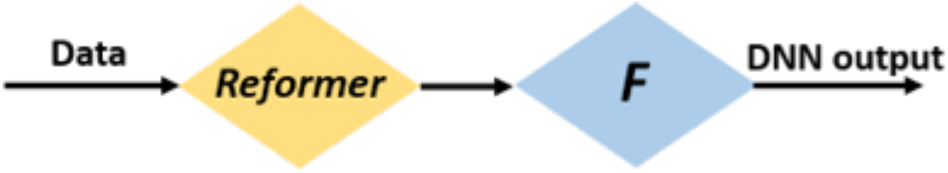


**Figure 5.3:** Input Reconstruction

## 5.3 State-of-the-art: Limitation Analysis

Through analysis, we find that the state-of-the-art defense mechanisms have intrinsic limitations for adversarial defense in control domain CPS applications due to different attack requirements and properties, as shown below:

### Adversarial Training

Adversarial training needs to generate adversarial examples for each batch of data during the training process, which increases the training computation overhead significantly. As demonstrated by line 8 in Algorithm 5, to avoid being removed by the detection scheme, the adversarial perturbations need to be projected to fit the constraint. The mapping process will further significantly introduce computation overhead to the adversarial training process. Therefore, adversarial training is not scalable to large systems that contain massive data resources.

### Adversarial Detection and Input Reconstruction

A common assumption of adversarial detection is that the adversarial examples follow a different distribution from normal inputs. The assumption is reasonable in the computer vision domain (the natural images will not contain the well-crafted perturbations) but not applicable for control domain CPS applications.

As introduced in chapter 4, the manifold of the normal measurements can be represented by the constraint $\Phi_{k\times r} M_C = \tilde{\Phi}$ empirically. To bypass the built-in detection of CPS application, the adversarial examples are also required to meet the constraint $\Phi_{k\times r} M_C^* = \tilde{\Phi}$. Intuitively, the crafted adversarial measurements share a similar manifold with the normal measurements. Therefore, adversarial detection will not work effectively in control domain CPS applications. This analysis can also be adapted to input reconstruction methods.

## 5.4 Random Input Padding Framework

As discussed above, the adversarial defense in control domain CPS application is non-trivial since the adversarial measurements share the same manifold as the normal measurements. Given the victim model, the attacker generates the perturbation iteratively through a gradient-based optimization process. As presented in [54], the perturbation generated by multi-step attacks usually has worse transferability, which indicates that adversarial perturbation is highly likely to be unique for each given data point in the adversarial attacks. Therefore, there is an intuition that the perturbation will no longer work if the input to the model changes. Inspired by the stochastic-based defense mechanisms in the computer vision field [72][107], we propose a random input padding defense framework to mitigate the effect of adversarial attacks in the control domain CPS applications.

The philosophy of our random input padding framework is straightforward, and the overall structure is shown in Fig 5.4. A random padding layer is added in front of the DNN in both training and inference stages. In general, the measurements of the sensors $\mathbf{z}$ are used as the features to train the detection models. Our framework firstly requires the operator to pick a padding dimension number $P$ ($P > m$) as the input feature numbers for the DNN. Thereafter, we pad $P - m$ zeros randomly to the plain inputs $\mathbf{z}$ and there will be $P - m$ padding scenarios in total. The DNN is then required to learn the pattern from the plain measurements that are embedded into the padded inputs during the training process. During the inference stage, when a new measurement vector $\mathbf{z}$ is received, the framework randomly pads $\mathbf{z}$ to a $P$ dimensional vector and feeds the padded vector to the DNN. Ideally, the detection rate against adversarial attacks should be $1 - \frac{1}{P-m+1}$ ($P \geq m$). The padding framework also works with possible input reshape, as shown in Fig 5.4.

As the padding process is random for each $\mathbf{z}$ at the inference stage, the attacker (and even the operator) cannot know the final DNN padded input vectors even when she/he knows the whole framework. For white-box attacks, the attacker will be able to generate perturbations for one of the $P - m$ padding scenarios. Since the multi-steps perturbations have relatively weak transferability, the adversarial attacks should have a lower success rate under the random padding framework. Intuitively, a larger $P$ will decrease the success rate

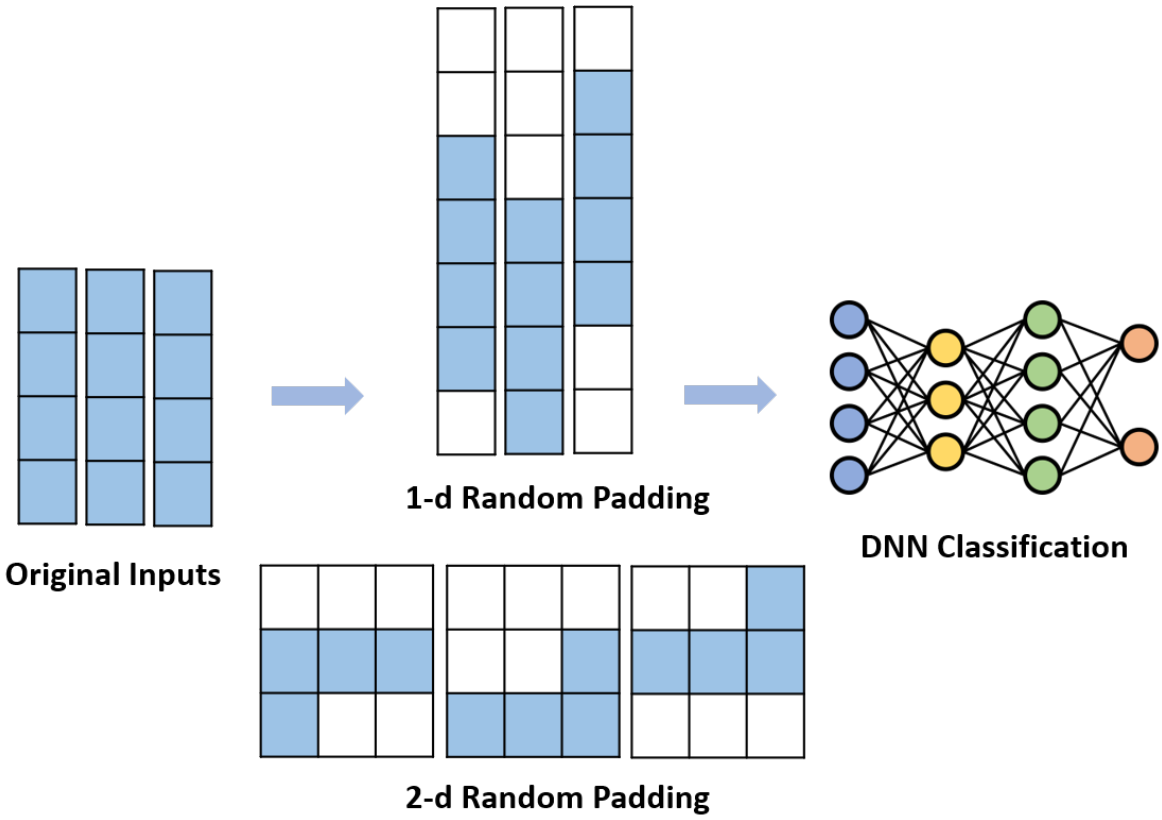


**Figure 5.4:** Illustration of random inputs padding framework

of adversarial attacks and finally increase the robustness of the DNN used for control domain CPS applications.

Different from [107], our framework requires input data pre-processing (padding) during the training stage and cannot be applied to a trained model directly. This is because the measurement data of a specific control domain CPS should follow the manifold defined by the physical property of the system, which will be destroyed if the measurement vectors are reshaped, resized, or sampled directly. Meanwhile, our framework only increases the computation of the training process slightly and is compatible with different neural networks.

## 5.5 Simulation Evaluation

### 5.5.1 Customer Domain CPS application: Energy Theft Detection

We analyze the properties of adversarial measurements and energy theft measurements and find that their distribution should be different. We then employ the t-Distributed Stochastic Neighbor Embedding (t-SNE) tool to reduce the measurement vectors into 2-d vectors and visualize their difference, as shown in Fig. 5.5.

From Fig. 5.5, we can learn that the manifolds of adversarial measurements and energy theft measurements are different, which matches the assumptions of adversarial detection and input reconstruction. We then generate a training dataset that contains 15,000 adversarial measurement vectors and the same number of energy theft measurement vectors. We use the dataset to train an auxiliary binary classifier DNN, and the DNN achieves over 98% classification accuracy. Therefore, adversarial detection can effectively distinguish adversarial measurements. For input reconstruction, we trained an FNN autoencoder with the normal energy theft measurements. After that, we feed the adversarial measurements to the autoencoder first and forward the output of the autoencoder to the energy theft detection DNN, the detection DNN then achieves over 97% detection accuracy.

We also evaluate the random padding framework, we train several LSTM DNNs with the same structure with $f_{RNN}$ in Table 3.3 except for the input dimension (match the padding).

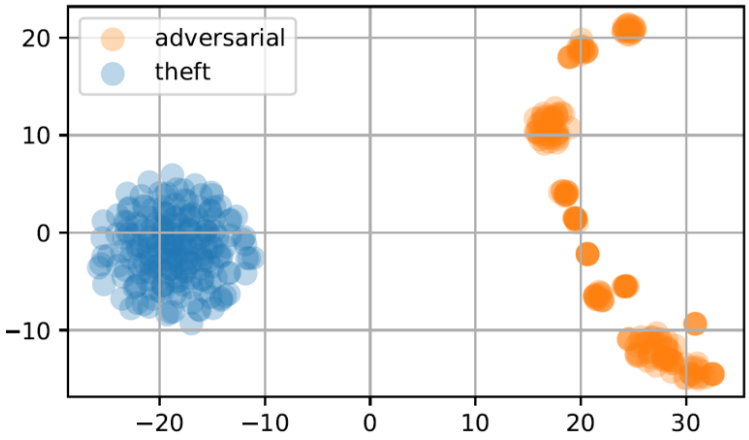


**Figure 5.5:** Energy theft/adversarial measurements visualization (t-SNE dimensionality reduction)

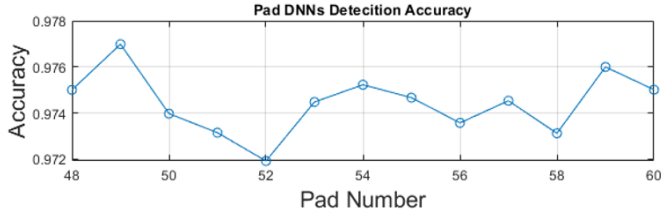


**Figure 5.6:** Detection recall of padded DNNs

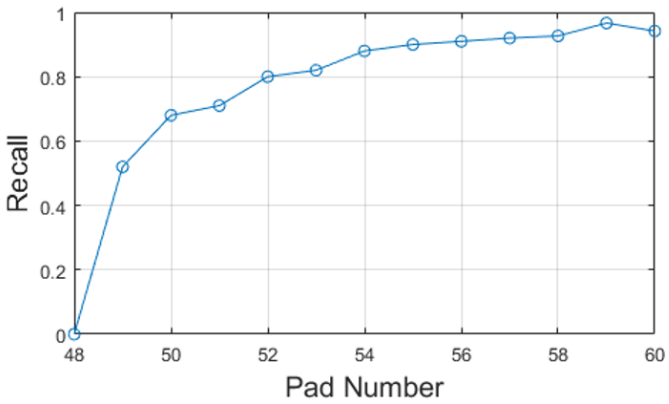


**Figure 5.7:** Detection recall of padded DNNs under adversarial attacks

We select LSTM DNNs in our simulation due to their detection performance. The evaluation results are demonstrated in Fig. 5.6 and Fig. 5.7. From Fig. 5.7, we can learn that the random padding framework significantly increases the robustness of DNNs under adversarial attacks. In addition, from Fig. 5.6, we can learn that the padding process will not influence the DNNs' detection performance on normal inputs.

### 5.5.2 Control Domain CPS application: FDIA detection

We employ the algorithm described in [54] to evaluate the performance of adversarial training in FDIA detection. We generate the adversarial measurements of all false records in each batch during the training process with the real-time trained DNN and added them to the training data. The training process takes 530 seconds to converge and achieves 98.5% overall detection accuracy and 99.2% detection recall of false measurements. For comparison, the normal training process takes around 75 seconds to converge. Meanwhile, we launch adversarial attacks to the adversarial trained DNN and the detection recall of false measurements decreases to 15.6%. Therefore, adversarial training is not appropriate for adversarial defense in FDIA detection.

We employ the adversarial detection methods described in [73] and generate the adversarial examples of all false measurements in the original training dataset. We use the original false measurements and their corresponding adversarial measurements to train a binary classification DNN $F^{adv}$. We empirically attempt different structures and parameters of the $F^{adv}$ and observe that its performance is not reliable. The best classification accuracy in our experiments is around 75%. As analyzed, we explain that this result is caused by the similar manifolds shared between the false measurements $\mathbf{z}_a$ and the adversarial measurements $\mathbf{z}_{adv}$. To verify our analysis, we utilize the t-SNE to visualize the manifolds in 2-dimensions, as shown in Fig. 5.8. From Fig. 5.8, we can learn that the adversarial measurements share a very similar manifolds with the normal false measurements. This is due to the the physical property of the power system and the constraints. Therefore, adversarial detection can not distinguish the adversarial measurements from model inputs effectively in FDIA detection.

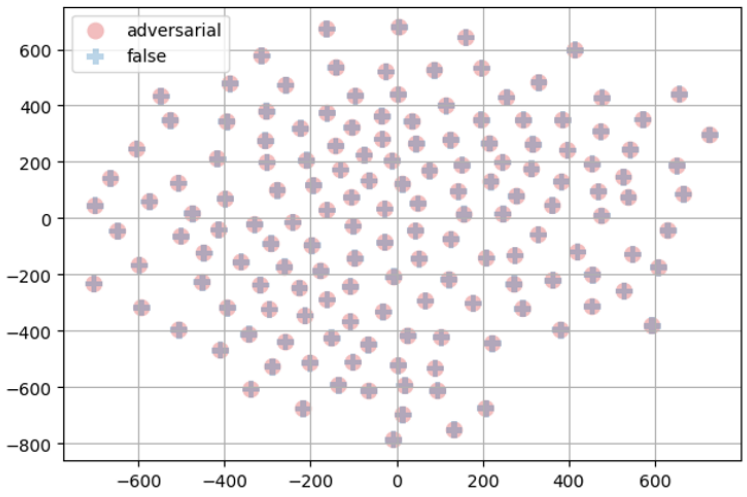


**Figure 5.8:** FDIA adversarial measurements visualization (t-SNE dimensionality reduction)

We utilize an autoencoder described in [72] and [38] to recover the general false measurements from their adversarial measurements. Similar to adversarial detection, the similar manifolds shared between the false measurements and adversarial measurements in FDIA decreases the effectiveness of input construction. A basic FNN is trained as the autoencoder $f_{encoder}$ with all records in $D_{train}$. We then evaluate the autoencoder loss of the false measurements and adversarial measurements described in [72]. As shown in Fig. 5.9, the loss of two kinds of measurement vectors is very close and difficult to separate.

We evaluate the defense performance of our random padding framework with three different types of DNNs, including FNN, long short-term memory (LSTM, acts as RNN), and CNN. We modify the number of the input neurons according to the padding number $P$ for all three DNNs. Meanwhile, we empirically select the kernel size of the CNN for different $P$ inputs in our experiments. The overall detection accuracy of three types of DNNs under the random padding framework is shown in Fig. 5.10. We can learn that with the padding number $P$ increases, the detection accuracy of FNN and CNN decreases gradually and becomes stable. The detection accuracy of FNN and CNN reaches aroung 96% and 95% respectively while RNN (LSTM) obtains a better performance (aorund 98%) under the random padding framework.

Figure 5.11 demonstrates the adversarial resistance property of our random padding framework. When there is no input padding ($P = 46$), all the models' detection performances drop to below 15%. However, with the increase of padding number $P$, the detection recall increases significantly and trends become stable to specific ranges. From the figure, we can learn that the FNN and RNN perform better than CNN in adversarial resistance. The best performance of FNN is 79.7% ($P = 70$), and for RNN and CNN is 89.5% ($P = 64$) and 71.8% ($P = 70$) respectively. Overall, our framework can remarkably increase the robustness of the DNNs in FDIA detection compared with previous state-of-the-art approaches.

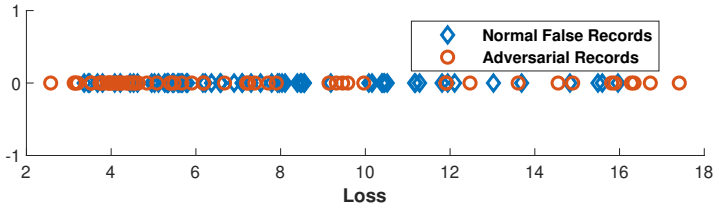


**Figure 5.9:** The autoencoder loss of false measurements and corresponding adversarial measurements

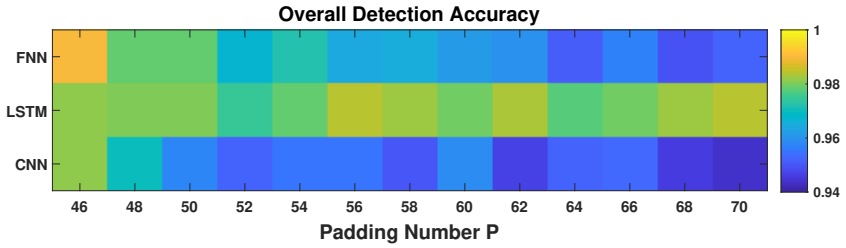


**Figure 5.10:** Detection accuracy of random inputs padding framework.

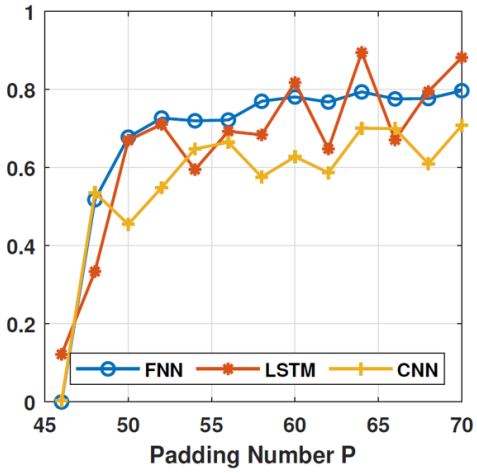


**Figure 5.11:** Detection recall of random inputs padding framework under adversarial attack.

# Chapter 6

# Conclusions and Future Works

In this dissertation, we investigate the potential security problems of employing deep learning techniques in cyber-physical system applications.

## 6.1 Conclusions

In chapter 2, we study the data privacy issues in cloud-assisted CPS data storage systems. We propose a practical searchable symmetric scheme that enables the user to query keywords from the encrypted ciphertext data. Compared with previous typical SSE methods, our scheme achieves high space-efficiency with little information disclosure that is tolerated for practical CPS applications.

In chapter 3, we study the adversarial machine learning in customer domain CPS applications with the DNN-based energy theft detection. We summarize the specific properties of the adversarial attacks and propose a practical threat model. We then propose the SearchFromFree framework which contains a random initialization scheme to maximize the attacker's profit and a step-size iterative scheme to increase the transferability of adversarial measurements. The evaluation based on a real-world smart meter dataset shows that our framework allows the adversarial attacker to report extremely low power consumption data to the utilities without being detected by the well-trained DNN models.

In chapter 4, we study adversarial machine learning in control domain CPS applications. From the attacker's perspective, we find that the control domain CPS applications propose

more challenges for the adversarial attacker. The main constraints are 1) knowledge constraints that prevent the attacker learning the measurements of uncompromised sensors and 2) physical constraint that requires the adversarial examples to follow the inner constraints defined by the physical system among the sensors. We then propose ConAML, a framework for adversarial attacks to control domain CPS applications. We evaluate the ConAML framework with three different applications and the result shows our framework enables the attacker to generate effective adversarial examples under practical constraints.

In chapter 5, we investigate the defense mechanisms of adversarial attacks. We evaluate the performance of several state-of-the-art defense mechanisms, including adversarial detection, adversarial training, and input reconstruction. However, we find that they have intrinsic limitations on defending against control domain adversarial attacks. To solve this, we propose a random padding framework to increase the robustness of DNNs. The evaluation based on both customer domain application (energy theft detection) and control domain application (FDIA detection) shows that our framework is resistant to white-box adversarial attack and outperforms the state-of-the-art approaches.

## 6.2 Future Research Directions

Based on the recent research on CPS security, we summarize the potential research directions:

- The SSE proposed in this dissertation can also be evaluated with other CPS applications whose data requires high-level privacy, such as medical CPSs.
- A more accurate and reliable deep learning-based incident detection model is needed. The current models can only indicate incidents of a highway, and more accurate models that indicate the specific incident location will be studied in the future.
- The adversarial examples are inevitable as long as the DNNs are not perfect. Therefore, more practical security solutions for practical CPS deep learning applications should be studied. An example of the preferred solution can be specific sensor protection schemes.

# Bibliography

[1] (2011). Ieee draft standard for synchrophasor data transfer for power systems. *IEEE PC37.118.2/D3.2, May 2011*, pages 1–54. 13

[2] (2018). Python Cryptography Toolkit. https://pypi.org/project/pycrypto/. [Online; accessed 16-Aug-2020]. 20

[3] (2018). Searchable Encryption. https://github.com/jiangnan3/Searchable_Encryption. [Online; accessed 16-Aug-2020]. 20

[4] Administration, U. E. I. (2016). Electricity. https://www.eia.gov/electricity. [Online; accessed 16-Mar-2020]. 18

[5] Ahmed, C. M., Zhou, J., and Mathur, A. P. (2018). Noise matters: Using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in cps. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 566–581. ACM. 43, 74

[6] Archive, I. S. S. D. (2012). CER Smart Metering Project. https://www.ucd.ie/issda/data/commissionforenergyregulationcer/. [Online; accessed 16-Mar-2020]. 30

[7] Arenas-Martínez, M., Herrero-Lopez, S., Sanchez, A., Williams, J. R., Roth, P., Hofmann, P., and Zeier, A. (2010). A comparative study of data storage and processing architectures for the smart grid. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 285–290. IEEE. 7

[8] Athay, T., Podmore, R., and Virmani, S. (1979). A practical method for the direct analysis of transient stability. *IEEE Transactions on Power Apparatus and Systems*, (2):573–584. x, 73

[9] Awareness, S. G. (2018). Hacking a Smart Meter and Killing the Grid. https://smartgridawareness.org/2018/10/27/killing-the-grid/. [Online; accessed 16-Mar-2020]. 22

[10] Ayad, A., Farag, H. E., Youssef, A., and El-Saadany, E. F. (2018). Detection of false data injection attacks in smart grids using recurrent neural networks. In *2018 IEEE*

*Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE. 1, 43, 71

[11] Bereş, A., Genge, B., and Kiss, I. (2015). A brief survey on smart grid data analysis in the cloud. *Procedia Technology*, 19:858–865. 3, 7

[12] Bi, S. and Zhang, Y. J. (2011). Defending mechanisms against false-data injection attacks in the power system state estimation. In *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, pages 1162–1167. IEEE. 71

[13] Bosch, C., Hartel, P., Jonker, W., and Peter, A. (2014). A survey of provably secure searchable encryption. *ACM Computing Surveys (CSUR)*, 47(2):1–51. 11

[14] Bost, R. (2016). $\sigma$ o$\varphi$o$\varsigma$: Forward secure searchable encryption. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1143–1154. 8, 10

[15] Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., and Zhou, W. (2016). Hidden voice commands. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 513–530. 44

[16] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., and Mukhopadhyay, D. (2018). Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*. 45

[17] Chang, Y.-C. and Mitzenmacher, M. (2005). Privacy preserving keyword searches on remote encrypted data. In *International Conference on Applied Cryptography and Network Security*, pages 442–455. Springer. 4, 8, 10, 17, 19

[18] Chen, Y., Poskitt, C. M., and Sun, J. (2018a). Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 648–660. IEEE. 43, 45, 74

[19] Chen, Y., Tan, Y., and Deka, D. (2018b). Is machine learning in power systems vulnerable? In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6. IEEE. 5, 23, 24

[20] Chen, Y., Tan, Y., and Zhang, B. (2019). Exploiting vulnerabilities of load forecasting through adversarial attacks. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, pages 1–11. 23, 25

[21] Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. (2014). The loss surface of multilayer networks. *CoRR*, abs/1412.0233. 58

[22] Curtmola, R., Garay, J., Kamara, S., and Ostrovsky, R. (2011). Searchable symmetric encryption: improved definitions and efficient constructions. *Journal of Computer Security*, 19(5):895–934. 4, 8, 10, 19

[23] Dabrowski, J. J., Rahman, A., George, A., Arnold, S., and McCulloch, J. (2018). State space models for forecasting water quality variables: an application in aquaculture prawn farming. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 177–185. ACM. 39

[24] Depuru, S. S. S. R., Wang, L., and Devabhaktuni, V. (2011). Support vector machine based data classification for detection of electricity theft. In *2011 IEEE/PES Power Systems Conference and Exposition*, pages 1–8. IEEE. 24

[25] Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. (2018). Boosting adversarial attacks with momentum. In *Proceedings of the IEEE CVPR*, pages 9185–9193. 39, 44

[26] Erba, A., Taormina, R., Galelli, S., Pogliani, M., Carminati, M., Zanero, S., and Tippenhauer, N. O. (2019). Real-time evasion attacks with physical constraints on deep learning-based anomaly detectors in industrial control systems. *arXiv preprint arXiv:1907.07487*. 44, 45

[27] Feng, C., Li, T., Zhu, Z., and Chana, D. (2017). A deep learning-based framework for conducting stealthy attacks in industrial control systems. *arXiv preprint arXiv:1709.06397*. 43, 74

[28] Feng, C., Palleti, V. R., Mathur, A., and Chana, D. (2019). A systematic framework to generate invariants for anomaly detection in industrial control systems. In *NDSS*. 43, 74

[29] for a Smarter Electric Grid, I. C. (2017). IEEE 39-Bus System. https://icseg.iti.illinois.edu/ieee-39-bus-system/. [Online; accessed 16-Aug-2020]. x, 73

[30] Gakis, E., Kehagias, D., and Tzovaras, D. (2014). Mining traffic data for road incidents detection. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 930–935. IEEE. 66

[31] Gao, Y., Foggo, B., and Yu, N. (2019). A physically inspired data-driven model for electricity theft detection with smart meter data. *IEEE Transactions on Industrial Informatics*, 15(9):5076–5088. 22, 30

[32] Ghafouri, A., Vorobeychik, Y., and Koutsoukos, X. (2018). Adversarial regression for detecting attacks in cyber-physical systems. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3769–3775. 44

[33] Goh, E.-J. (2004). Secure indexes. 4, 8, 10, 13, 14, 17, 19

[34] Goh, J., Adepu, S., Junejo, K. N., and Mathur, A. (2016). A dataset to support research in the design of secure water treatment systems. In *International Conference on Critical Information Infrastructures Security*, pages 88–99. Springer. 74

[35] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*. 23, 24, 27, 39, 40, 43, 53, 56

[36] Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE. 39

[37] Grosse, K., Papernot, N., Manoharan, P., Backes, M., and McDaniel, P. (2017). Adversarial examples for malware detection. In *European Symposium on Research in Computer Security*, pages 62–79. Springer. 44

[38] Gu, S. and Rigazio, L. (2014). Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*. 82, 91

[39] Hahn, F. and Kerschbaum, F. (2014). Searchable encryption with secure and efficient updates. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 310–320. 8, 10

[40] Han, X. (2019). Traffic incident detection: A deep learning framework. In *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, pages 379–380. IEEE. 1, 43, 65, 66

[41] Hart, G. W. (1992). Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891. 8

[42] Hasan, M., Toma, R. N., Nahid, A.-A., Islam, M., Kim, J.-M., et al. (2019). Electricity theft detection in smart grid systems: a cnn-lstm based approach. *Energies*, 12(17):3310. 22, 24

[43] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. 39

[44] He, Y., Mendis, G. J., and Wei, J. (2017). Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism. *IEEE Transactions on Smart Grid*, 8(5):2505–2516. 1, 43, 71

[45] Inoue, J., Yamagata, Y., Chen, Y., Poskitt, C. M., and Sun, J. (2017). Anomaly detection for a water treatment system using unsupervised machine learning. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1058–1065. IEEE. 43, 74

[46] Ismail, M., Shaaban, M. F., Naidu, M., and Serpedin, E. (2020). Deep learning detection of electricity theft cyber-attacks in renewable distributed generation. *IEEE Transactions on Smart Grid*. 4, 22, 24

[47] James, J., Hou, Y., and Li, V. O. (2018). Online false data injection attack detection with wavelet transform and deep neural networks. *IEEE Transactions on Industrial Informatics*, 14(7):3271–3280. 1, 43, 71

[48] Jindal, A., Dua, A., Kaur, K., Singh, M., Kumar, N., and Mishra, S. (2016). Decision tree and svm-based data analytics for theft detection in smart grid. *IEEE Transactions on Industrial Informatics*, 12(3):1005–1016. 24

[49] Jokar, P., Arianpoo, N., and Leung, V. C. (2015). Electricity theft detection in ami using customers' consumption patterns. *IEEE Transactions on Smart Grid*, 7(1):216–226. 1, 22, 24, 30

[50] Kamara, S., Papamanthou, C., and Roeder, T. (2012). Dynamic searchable symmetric encryption. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 965–976. 8, 10

[51] Kong, W., Dong, Z. Y., Hill, D. J., Luo, F., and Xu, Y. (2017). Short-term residential load forecasting based on resident behaviour learning. *IEEE Transactions on Power Systems*, 33(1):1087–1088. 1

[52] korba, A. A. (2018). Energy fraud detection in advanced metering infrastructure ami. In *Proceedings of the 7th International Conference on Software Engineering and New Technologies*, pages 1–6. 1, 22, 24, 30

[53] Kravchik, M. and Shabtai, A. (2018). Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, pages 72–83. ACM. 39, 43, 74

[54] Kurakin, A., Goodfellow, I., and Bengio, S. (2016a). Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*. 27, 44, 82, 85, 89

[55] Kurakin, A., Goodfellow, I., and Bengio, S. (2016b). Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*. 39, 43, 44, 53

[56] Labs, I. (2019). Secure Water Treatment (SWaT) Dataset. https://itrust.sutd.edu.sg/itrust-labs_datasets. [Online; accessed 15-08-2019]. 74

[57] Li, H., Yang, Y., Dai, Y., Bai, J., Yu, S., and Xiang, Y. (2017). Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data. *IEEE Transactions on Cloud Computing*. 11

[58] Li, J., Ji, S., Du, T., Li, B., and Wang, T. (2018). Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*. 44

[59] Li, J., Lee, J. Y., Yang, Y., Sun, J. S., and Tomsovic, K. (2020a). Conaml: Constrained adversarial machine learning for cyber-physical systems. *arXiv preprint arXiv:2003.05631*. 23, 25

[60] Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., and Lou, W. (2010). Fuzzy keyword search over encrypted data in cloud computing. In *2010 Proceedings IEEE INFOCOM*, pages 1–5. IEEE. 8, 10

[61] Li, L., Liang, C.-J. M., Liu, J., Nath, S., Terzis, A., and Faloutsos, C. (2011). Thermocast: A cyber-physical forecasting model for datacenters. In *Proceedings of the 17th ACM SIGKDD*, KDD '11, pages 1370–1378. ACM. 39

[62] Li, L., Lin, Y., Du, B., Yang, F., and Ran, B. (2020b). Real-time traffic incident detection based on a hybrid deep learning model. *Transportmetrica A: Transport Science*, pages 1–21. 1, 43, 65

[63] Liang, G., Weller, S. R., Zhao, J., Luo, F., and Dong, Z. Y. (2016). The 2015 ukraine blackout: Implications for false data injection attacks. *IEEE Transactions on Power Systems*, 32(4):3317–3318. 67, 82

[64] Liu, L. and Chen, R.-C. (2017). A novel passenger flow prediction model using deep learning methods. *Transportation Research Part C: Emerging Technologies*, 84:74–91. 65

[65] Liu, T. and Shu, T. (2019). Adversarial false data injection attack against nonlinear ac state estimation with ann in smart grid. In *International Conference on Security and Privacy in Communication Systems*, pages 365–379. Springer. 5, 25

[66] Liu, Y., Ning, P., and Reiter, M. K. (2009). False data injection attacks against state estimation in electric power grids. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 21–32. 70

[67] Lu, J., Sibai, H., Fabry, E., and Forsyth, D. (2017). No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*. 44

[68] Lu, Y. (2012). Privacy-preserving logarithmic-time search on encrypted data in cloud. In *NDSS*. 8, 10

[69] Ma, S. and Liu, Y. (2019). Nic: Detecting adversarial samples with neural network invariant checking. In *Proceedings of the 26th NDSS*. 82

[70] Maamar, A. and Benahmed, K. (2018). Machine learning techniques for energy theft detection in ami. In *Proceedings of the 2018 International Conference on Software Engineering and Information Management*, pages 57–62. 4

[71] Marulli, F. and Visaggio, C. A. (2019). Adversarial deep learning for energy management in buildings. In *Proceedings of the 2019 Summer Simulation Conference*, page 50. Society for Computer Simulation International. 25

[72] Meng, D. and Chen, H. (2017). Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM CCS*, pages 135–147. 82, 85, 91

[73] Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. (2017). On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*. 82, 89

[74] Miers, I. and Mohassel, P. (2017). Io-dsse: Scaling dynamic searchable encryption to millions of indexes by improving locality. In *NDSS*. 8, 10

[75] Moosavi-Dezfooli, S., Fawzi, A., and Frossard, P. (2016). Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582. 39, 43, 53, 60

[76] Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. (2017). Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773. 39, 44, 53, 56

[77] Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE CVPR*, pages 2574–2582. 23, 24, 27

[78] Nabil, M., Ismail, M., Mahmoud, M., Shahin, M., Qaraqe, K., and Serpedin, E. (2018). Deep recurrent electricity theft detection in ami networks with random tuning of hyper-parameters. In *2018 ICPR*, pages 740–745. IEEE. 1, 4, 22, 24, 30

[79] Nagi, J., Yap, K. S., Tiong, S. K., Ahmed, S. K., and Mohamad, M. (2009). Nontechnical loss detection for metered customers in power utility using support vector machines. *IEEE transactions on Power Delivery*, 25(2):1162–1171. 24

[80] Nagi, J., Yap, K. S., Tiong, S. K., Ahmed, S. K., and Nagi, F. (2011). Improving svm-based nontechnical loss detection in power utility using the fuzzy inference system. *IEEE Transactions on power delivery*, 26(2):1284–1285. 24

[81] Niu, X., Li, J., Sun, J., and Tomsovic, K. (2019). Dynamic detection of false data injection attack in smart grid using deep learning. In *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–6. IEEE. 1, 43, 71

[82] of Transportation, C. D. (2021). Caltrans Performance Measurement System (PeMS). http://pems.dot.ca.gov/. [Online; accessed 26-Mar-2021]. 66

[83] Ozay, M., Esnaola, I., Vural, F. T. Y., Kulkarni, S. R., and Poor, H. V. (2015). Machine learning methods for attack detection in the smart grid. *IEEE transactions on neural networks and learning systems*, 27(8):1773–1786. 1, 39, 43, 71

[84] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016). The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE. 61

[85] Punmiya, R. and Choe, S. (2019). Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. *IEEE Transactions on Smart Grid*, 10(2):2326–2329. 22, 30

[86] Ren, H., Song, Y., Wang, J., Hu, Y., and Lei, J. (2018). A deep learning approach to the citywide traffic accident risk prediction. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3346–3351. IEEE. 1, 43, 65

[87] Ren, J. S., Wang, W., Wang, J., and Liao, S. (2012). An unsupervised feature learning approach to improve automatic incident detection. In *2012 15th International IEEE conference on intelligent transportation systems*, pages 172–177. IEEE. 1, 43, 65

[88] Rndic, N. and Laskov, P. (2014). Practical evasion of a learning-based classifier: A case study. In *2014 IEEE symposium on security and privacy*, pages 197–211. IEEE. 44, 45

[89] Rozsa, A., Rudd, E. M., and Boult, T. E. (2016). Adversarial diversity and hard positive generation. In *Proceedings of the IEEE CVPR Workshops*, pages 25–32. 23, 24, 27, 39, 43, 53, 56, 61

[90] Rusitschka, S., Eger, K., and Gerdes, C. (2010). Smart grid data cloud: A model for utilizing cloud computing in the smart grid domain. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 483–488. IEEE. 7

[91] Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. (2019). Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3353–3364. 82

[92] Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. (2016). Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540. ACM. 44

[93] Shi, H., Xu, M., and Li, R. (2017). Deep learning for household load forecasting—a novel pooling deep rnn. *IEEE Transactions on Smart Grid*, 9(5):5271–5280. 1

[94] Song, D. X., Wagner, D., and Perrig, A. (2000). Practical techniques for searches on encrypted data. In *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000*, pages 44–55. IEEE. 4, 8, 9, 19

[95] Sun, T. (2017). ELECTRIC SMART METER HACK. https://www.youtube.com/watch?v=CjrJjMNrqsI. [Online; accessed 16-Mar-2020]. 22

[96] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*. 23, 24, 39, 43

[97] Tan, S., De, D., Song, W.-Z., Yang, J., and Das, S. K. (2017). Survey of security advances in smart grid: A data driven approach. *IEEE Communications Surveys & Tutorials*, 19(1):397–422. 1

[98] Ten, C.-W., Liu, C.-C., and Manimaran, G. (2008). Vulnerability assessment of cybersecurity for scada systems. *IEEE Transactions on Power Systems*, 23(4):1836–1846. 67

[99] Tian, Y., Pei, K., Jana, S., and Ray, B. (2018a). Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th international conference on software engineering*, pages 303–314. ACM. 44

[100] Tian, Y., Zhang, K., Li, J., Lin, X., and Yang, B. (2018b). Lstm-based traffic flow prediction with missing data. *Neurocomputing*, 318:297–305. 65

[101] Tong, Y. (2015). *Data security and privacy in smart grid*. PhD thesis, University of Tennessee, Knoxville. 67

[102] Tong, Y., Deyton, J., Sun, J., and Li, F. (2013a). $s^3a$: A secure data sharing mechanism for situational awareness in the power grid. *IEEE Transactions on Smart Grid*, 4(4):1751–1759. 10, 13

[103] Tong, Y., Sun, J., Chow, S. S., and Li, P. (2013b). Cloud-assisted mobile-access of health data with privacy and auditability. *IEEE Journal of biomedical and health Informatics*, 18(2):419–429. 10

[104] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. (2017). Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*. 81, 82

[105] Wang, C., Tindemans, S., Pan, K., and Palensky, P. (2020). Detection of false data injection attacks using the autoencoder approach. *arXiv preprint arXiv:2003.02229*. 1, 43, 71

[106] Wood, A. J., Wollenberg, B. F., and Sheblé, G. B. (2013). *Power generation, operation, and control*. John Wiley & Sons. 40

[107] Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. (2018). Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*. 85, 87

[108] Xu, H., Ma, Y., Liu, H.-C., Deb, D., Liu, H., Tang, J.-L., and Jain, A. K. (2020). Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2):151–178. 81

[109] Xu, W., Evans, D., and Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*. 82

[110] Xu, W., Qi, Y., and Evans, D. (2016). Automatically evading classifiers. In *Proceedings of the 2016 Network and Distributed Systems Symposium*, pages 21–24. 44

[111] Yan, J., Tang, B., and He, H. (2016). Detection of false data attacks in smart grid with supervised learning. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1395–1402. IEEE. 1, 43, 71

[112] Yang, Q., An, D., Min, R., Yu, W., Yang, X., and Zhao, W. (2017). On optimal pmu placement-based defense against data integrity attacks in smart grid. *IEEE Transactions on Information Forensics and Security*, 12(7):1735–1750. 71

[113] Yuan, X., He, P., Zhu, Q., and Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*. 45

[114] Yuan, Z., Lu, Y., Wang, Z., and Xue, Y. (2014). Droid-sec: deep learning in android malware detection. In *ACM SIGCOMM Computer Communication Review*, volume 44, pages 371–372. ACM. 39

[115] Yuan, Z., Zhou, X., and Yang, T. (2018). Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 984–992. 1, 43, 65, 66

[116] Zanetti, M., Jamhour, E., Pellenz, M., Penna, M., Zambenedetti, V., and Chueiri, I. (2017). A tunable fraud detection system for advanced metering infrastructure using short-lived patterns. *IEEE Transactions on Smart Grid*, 10(1):830–840. ix, 1, 22, 30, 31

[117] Zhang, K., Zheng, L., Liu, Z., and Jia, N. (2020). A deep learning based multitask model for network-wide traffic speed prediction. *Neurocomputing*, 396:438–450. 65

[118] Zheng, K., Chen, Q., Wang, Y., Kang, C., and Xia, Q. (2018). A novel combined data-driven approach for electricity theft detection. *IEEE Transactions on Industrial Informatics*, 15(3):1809–1819. 22, 30

[119] Zheng, Z., Yang, Y., Niu, X., Dai, H.-N., and Zhou, Y. (2017). Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. *IEEE Transactions on Industrial Informatics*, 14(4):1606–1615. 1, 4, 22, 24

[120] Zhu, L., Guo, F., Krishnan, R., and Polak, J. W. (2018). A deep learning approach for traffic incident detection in urban networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1011–1016. IEEE. 1, 43, 65

[121] Zimmerman, R. D., Murillo-Sánchez, C. E., and Thomas, R. J. (2010). Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on power systems*, 26(1):12–19. 72

# Vita

Jiangnan Li was born in Weinan, China. He received a B.S. degree from the University of Electronic Science and Technology of China in 2016. He is pursuing a Ph.D. degree in the Department of Electrical Engineering and Computer Science (EECS) at the University of Tennessee, Knoxville, and is currently a research assistant under the supervision of Dr. Jinyuan Sun. His research interests include applied cryptography, deep learning security, cyber-physical system security, and their overlaps.