IMPERIAL

Blending Data and Model for Robust and Secure Power System Operation

by

Wangkun Xu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

at
Imperial College London
2024

Blending Data and Model for Robust and Secure Power System Operation

Wangkun Xu

Abstract

The evolving landscape of power systems, characterized by the trend of *decarbonization*, *digitalization* and *decentralization*, demands more efficient, robust, and secure operation strategies. Traditional *model-based* approaches are being challenged, leading to a transition to *data-driven* methods enabled by advances in information and communication technologies. However, concerns persist regarding the interpretability and reliability of purely data-driven decision-making processes. Hence, this thesis explores an intermediate approach that blends the data and the model for power system operation, offering a viable solution to the new challenges.

Two distinct frameworks are examined, each offering varying degrees of integration. The first framework orchestrates sequential learning and optimization processes to facilitate the exchange of critical information. The second framework embeds optimization models within deep learning structures, enabling the forecast to be decision-aware.

Chapter 2 presents a robust moving target defence method for the detection of false data injection attacks. By optimizing the set points of distributed flexible AC transmission system devices in real-time, the method maximizes the detection probability under specific measurement noise levels. Within the context of sequential design in Chapter 3, the thesis illustrates how a data-driven attack detector and physics-informed attack identifier can spatially and temporally reduce the operational cost of robust moving target defence by quantifying its uncertainty set. The sequential design instills greater trust among system operators, compared to its pure data-driven counterpart.

Chapter 4 evaluates the generalizability of the integrated framework. A unified adversarial training approach is proposed to address its uncertainties in both the input space of the deep neural network and the parameter spaces of model-based optimization. In Chapter 5, the integrated framework is introduced to facilitate machine unlearning tasks in load forecasting, providing a balance between data privacy and the operation cost of the whole system.

天行健，君子以自强不息；地势坤，君子以厚德载物。
——《周易》

As heaven maintains vigor through movements,
A gentle man should constantly strive for self-perfection;
As earth's condition is receptive devotion,
A gentle man should hold the outer world with broad mind.
'The Book of Changes'

To my wife, my family, and my cats.

Acknowledgements

I extend my deepest gratitude to those who served as unwavering pillars of support throughout this challenging 4-year Ph.D. journey.

To my family, your boundless love and encouragement sustained me through the highs and lows. Your belief in my abilities fueled my determination, and to my wife, Ms. Shuhang Pang, I express heartfelt thanks for her indispensable help and support. Special mention goes to my two lovely cats, Yoda and Kiwi, whose constant distraction provided moments of respite from my research pursuits.

A profound appreciation is reserved for my academic supervisors, Dr. Fei Teng and Dr. Imad Jaimoukha, whose guidance has greatly shaped my research and scholarly endeavors. Their wisdom, patience, and commitment to excellence played a pivotal role in my academic growth.

This work is also dedicated to all colleagues in the Control and Power research group at Imperial College London. I extend my appreciation to esteemed senior colleagues, Dr. Martin Higgins, Dr. Zhongda Chu, and Dr. Pudong Ge, for their generous contributions in providing constructive suggestions and engaging in numerous discussions pertaining to my research. Dr. Jianhong Wang, Dr. Jianli Gao, Mr. Junyu Mao, and Mr. Han Bai, your shared joys of discovery and collective perseverance through setbacks made the academic voyage more enjoyable. I extend special thanks to Dr. Zihang Dong, Dr. Cheng Hu, and Dr. Tingqi Zhang for their invaluable support during the initial stages of my Ph.D. research and for the delightful moments spent discussing football, adding an extra layer of joyness to my academic journey. Lastly, I wish to express my gratitude to Arsenal Football Club, a source of enthusiasm and inspiration for the past 15 years. Their spirit of resilience and determination has taught me invaluable lessons about perseverance and never giving up.

Finally, I dedicate this thesis to the countless individuals who paved the way for knowledge and progress. May this work contribute, even in a small way, to the collective pursuit of understanding and making a positive impact on the world.

With heartfelt appreciation,

Wangkun Xu
April, 2024 at London home

Statement of Originality

I, Wangkun Xu, declare that the material contained within this thesis is my own work, except where other work is appropriately referenced. Any use of the first-person plural such as “we” and “our” is for the sake of clarity.

Copyright Declaration

Nomenclature

Abbreviations

BDD	Bad Data Detection
cdf	Cumulative Density Function
CNN	Convolutional Neural Network
CO	Constrained Optimization
CPPS	Cyber-Physical Power System
CPSG	Cyber-Physical Smart Grid
D-FACTS	Distributed Flexible AC Transmission System
DDET-MTD	Data-Driven Event-Triggered MTD
DL	Deep Learning
DoF	Degree of Freedom
E2E	End-to-end
E2E-AT	End-to-end Adversarial Training
ED	Economic Dispatch
FDI	False Data Injection
FPR	False Positive Rate
ICT	Information and Communication Technique
LMI	Linear Matrix Inequality
LP	Linear Programming
LSTM-AE	Long Short-term Memory AutoEncoder
MAPE	Mean Absolute Percentage Error
MILP	Mixed Integer Linear Programming

ML	Machine Learning
MLP	Multi Layer Perceptron
MSE	Mean Squared Error
MTD	Moving Target Defence
MU	Machine Unlearning
NN	Neural Network
OPF	Optimal Power Flow
OR	Operation Research
PA/TA-MU	Performance/Task Aware Machine Unlearning
pdf	Probability Density Function
PGD	Projected Gradient Descent
PMU	Phaser Measurement Unit
PSSE	Power System State Estimation
QP	Quadrtic Programming
RES	Renewable Energy Resource
ROC	Receiver Operating Characteristic
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SE	State Estimation
SGD	Stochastic Gradient Descent
SO	System Operator
SQP	Sequential Quadratic Programming
TPR	True Positive Rate
UC	Unit Commitment

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The power system is undergoing a profound transformation, shifting from a fossil-fuel-based paradigm to one characterized by the widespread integration of renewable energy sources (RESes) [1]. This transition is accompanied by the emergence of decentralization and digitalization, fundamentally reshaping the operational framework of the power system. In detail, with the requirement of net-zero, the increased integration of renewable energy presents both opportunities and challenges for the operation of the power system. First, the intrinsic variability of renewable energy sources requires a more robust and efficient operation and market strategy, which poses a substantial challenge to conventional *model-based optimization method*. Second, massive penetration of distributed energy sources (DERs) requires redesigning the operation algorithm in a decentralized manner. Finally, as more DERs and increasingly complex communication systems are interconnected with the grid, the power system is evolving into a cyber-physical entity, susceptible to malicious activities.

In response to these challenges, *data-driven approaches*, notably *deep learning (DL)*, have been explored to extend the capabilities of traditional model-based methods, owing to their exceptional ability to represent unknown data patterns. The adoption of data-driven approaches empowers the system operator (SO) to simulate the intricate dynamics of the grid, even in the presence of limited knowledge about the underlying mechanisms. Moreover, data-driven algorithms facilitate the relocation of most online computation efforts, inherent in model-based counterparts, to the offline training phase, offering promising prospects for real-time operations in complex power systems.

The emerging need for a data-driven approach meets the blossom of advanced measurement systems, communication, and digitalization of power systems, such as smart meters and phasor measurement units (PMUs). The feasibility of training complex learning algorithms has become more pronounced with the substantial accumulation of publicly available data facilitated by these advanced measurement systems. In addition, advances in computing hardware have significantly alleviated the computational burden associated with training deep neural networks.

In fact, data-driven methods, such as deep learning, have seen active research and application in various areas of the power system. Note that the purpose of applying data-driven approaches is either to expedite online computation or to implicitly capture the underlying dynamic of the grid, which may be challenging to model analytically. Here is a brief overview based on various categories of data-driven algorithms:

- **Regression**. The most practical field of DL for the power system is forecasting, where a regression model is trained to forecast the future load or renewable energy [2]. Other regression tasks include predicting the trajectory of the rotor angle for preventive redispatch [3], learning for optimal power flow [4, 5] and state estimation [6], etc.

- **Classification**. Security assessment is an example of binary classification task in power system. The frequency and angle stability are classified to inform a redispatch action [7] or can be used as a preventive constraint in the dispatch stage [8]. In addition, anomaly detection and localization have attracted great attention due to the increasing level of digitalization and decentralization [9]. More detailed reviews on this topic are provided in Section 1.5.1.

- **Generative Model**. Generative models can be used to enrich the historical dataset.

For example, new renewable energy scenarios can be generated for better grid planning and operation [10]. Missing data can also be imputed via generative model in low-observable applications [11].

- **Reinforcement Learning**. Unlike the aforementioned DL techniques, reinforcement learning studies the interaction between the operator and the power grid. Therefore, the objective is to learn the optimal operation and control actions for dynamic power grid, such as energy dispatch [12], energy market [13], and voltage control [14].

These data-driven techniques play a crucial role in advancing various aspects of power system planning, operation, and control, contributing to the transition toward more efficient, reliable, and resilient grid.

Declaration on the Terminologies

In this thesis, two decision-making frameworks are discussed. Without introducing ambiguity, the term “data-driven algorithm/approach/method”, mainly referring to the deep learning algorithm, denotes the direct learning of decisions from the historic dataset, while “model-based optimization/approach/method” pertains to the design of policies based on existing models, usually via optimization programming. Due to the convention in DL community, the thesis also uses the term “model” in “deep learning model”, “neural network model” or “parametric model”, which should be distinguished by the reader from the term “model-based”.

1.2 Motivations and Research Questions

This section starts by shedding light on the constraints of data-driven algorithms, which emerge as a significant hurdle to their practical implementation. While there is a consensus on the potential of data-driven approaches for powering the future intelligent net-zero power system, this thesis contends that an intermediary phase is necessary, one that integrates the insights from models with the advancements made in data-driven algorithms. Based on the analysis, this section further motivates the objective of the thesis and proposes the research questions which are intended to be answered.

Regardless of its great success in vision and language tasks, the black-box nature of deep learning makes it difficult for the SO to accept it immediately for three reasons.

1. **Lack of Explainability**. Most recent deep learning algorithms are modeled by neural networks (NNs) whose representation ability can only be partially explained. Therefore, its reliability cannot be fully understood and trusted by the system operator that oversees the critical infrastructure.

2. **Lack of Proof of Generalizability**. The well-known generalization error caused by the out-of-distribution samples implies that the performance of the DL models can hardly be guaranteed in unknown scenarios. For instance, how can system operators be sure that their collected load consumption data in a certain time window are also representative to the future? For security-constrained physical power systems, the lack of generalization not only results in the reduction of economic efficiency, but also causes system failures, such as blackouts, which should always be avoided under any circumstance.

3. **Ethics, Security, and Privacy Concerns**. Attention must be directed toward the ethical and security considerations associated with deep learning algorithms. In particular, DL researchers demonstrate that small, specifically designed perturbations of neural network input could lead to significant accuracy drops, as highlighted by the concept of adversarial attacks [15]. In particular, this NN defect has not been fully resolved for the past ten years, which exposes potential security risks and challenges the applicability of neural networks in critical infrastructures. In addition, the collection of consumer electricity data can raise new concerns about possible privacy violations.

In contrast, the model-based approach, such as model-based operation and control relying on optimization programming, has a well-developed theory to support its optimality and robustness. In such a method, a systematic model is usually required to be built with reasonable assumptions and prior knowledge. Although not as flexible as its data-driven counterpart, the model-based approach often guarantees worst-case performance locally under extreme conditions. This presents a clear advantage over the data-driven approach, enabling system operators to comprehend the limitations of their algorithm and design compensation principles as backup plans.

Despite the drawbacks associated with data-driven decision-making, an additional argument posits that neglecting the well-developed physical model of the power system is wasteful and regrettable. Numerous existing power system modeling and operation techniques have been successfully built and verified over decades in real-time applications. Even in the face of a high penetration of renewable energy and the emergence of digitization, model-based approaches remain reliable and adept for various tasks.

Based on the above discussions on the interconnections between model-based and data-driven methods, the main goal of this thesis is to answer the following questions:

- *What are the advantages of co-designing a data-driven and model-based algorithm on both sides?*
- *If it has benefits, what are the viable options/frameworks to design such algorithm?*
- *How to demonstrate the performance of blending data and model for power system operations?*

This thesis aims to address these questions by developing theoretical sound algorithms and validating their performance in real-world power system operation tasks. Meanwhile, it is intended to show that the expertise in system identification and modeling is instructive and meaningful for designing accurate and reliable deep learning algorithms. Meanwhile, by leveraging the latest analytical results from data-driven techniques, the optimality and robustness of the model-based algorithm can be improved. By all means, this thesis does not assert that purely automatic operation cannot be achieved; rather, it emphasizes that

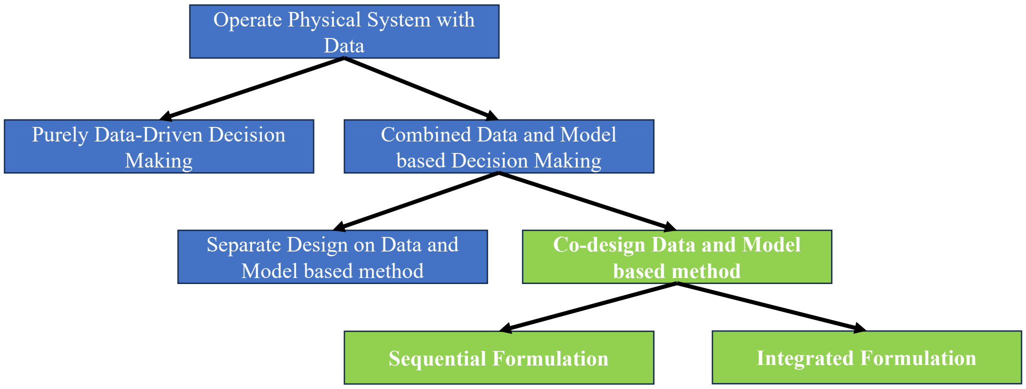

Operate Physical System with Data
Purely Data-Driven Decision Making
Combined Data and Model based Decision Making
Separate Design on Data and Model based method
Co-design Data and Model based method
Sequential Formulation
Integrated Formulation

Figure 1.1: The taxonomy of decision makings on real-world physical system with data. Interests of this thesis are highlighted in green.

Table 1.1: Comparison on frameworks of decision making with data.

Framework	**Training Pipeline**	**DL Model**	**Training Objective**
Purely Data-Driven Decision Making	Train on the decision of the down-stream task(s)	Single DL model for both forecast and decision-making	Objective of downstream task(s)
Separate Design on Data and Model	Train on the forecaster	DL Forecaster	Forecast error
Sequential Formulation	Train on the forecaster with indirect information of downstream task(s)	DL forecaster	Forecast error
Integrated Formulation (End-to-end learning)	Train on the forecaster with downstream task(s) directly embedded	DL forecaster	Objective of downstream task(s)

emerging technique requires time and effort to integrate into the existing system before potentially replacing it.

1.3 Introduction to Decision Makings with Data

The great representation capabilities of DL models, evident in their universal approximation property, have led to successful applications across various practical domains, including large language model and computer vision. Continuous advances in DL algorithms have promising prospects for improving the operations of critical industrial facilities, such as power systems. In contrast to applications in language and vision tasks, industrial processes are characterized by a sequence of decision-making tasks. These processes invariably involve predictable elements that are unknown at the time of decision-making.

From the perspective of different automation levels, the deployment of DL models for power system operation can be classified into purely and partially data-driven approaches. The taxonomy of data and model-based decision-makings are reported in Fig.1.1 depending on their different structures of the training pipeline, different types of parametric forecast model, and the training objective used during training [16]. A brief comparison is summarized in Table 1.1.

In a purely data-driven approach, the system operator uses DL models alone to perform a broad task of its entire process cycle with little human interference. This so-called *decision rule optimization* intends to find a parametric mapping from the available data to the optimal decision rule. For a dynamic system, it shares similarity with reinforcement learning. The inference and training procedure for the purely data-driven approach is highlighted in Fig.1.2. The benefit of the method is that it is extremely efficient during the inference time, as the decision can be generated directly by the forecast model. In this scope, an automated agent can be built into the power system to first implicitly forecast load consumption and renewable generation based on contextual information and then automatically dispatch energy into the system while controlling any potential contingencies.

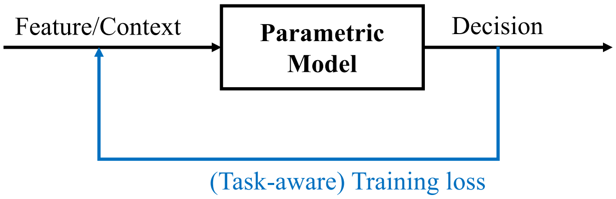


Figure 1.2: The inference and training (in blue) procedure of purely data-driven decision making.

Due to its black-box nature, currently the SO may be more willing to use DL models partially in the sequence of complete decision-making. For example, compared to direct learning on operational strategies, DL-based algorithms can be used more reliably for forecasting tasks, such as load forecasting in power system operation, demand forecasting in retail, and inventory stock forecasting in commerce. Downstream tasks, such as generator dispatch, are still run by humans using conventional model-based approaches. Although training such DL models is straightforwardly a supervised learning problem, statistical training criteria, such as mean squared error, may not be aligned with the ultimate goal of downstream decision-making tasks. Although a perfect prediction would always result in optimal decisions, the forecast error is inevitable and can propagate, especially on the test dataset. Physics-informed learning is another example of how physical models can be used to guide the training of deep neural networks. By embedding universal rules, such as algebraic and differentiable equations, the DL model can explicitly respect the physical constraints of the underlying dynamic [17]. Furthermore, all decision-makings are subject to uncertainties. Conventional optimization models ignore contextual information and apply unconditional distributions of the uncertain parameter to make decisions. Such decisions can be biased to the real-time situation, causing violations of its constraints and infeasible decisions. The formulation is usually conservative, and the objective becomes suboptimal due to the large uncertainties on its parameter space. Therefore, the modeling of contextual information such as prior knowledge and observations by DL can be used as a prescription for model-based optimizations.

From the above arguments, instead of treating individual tasks in the entire decision making separately using distinct data and model based approaches, blending data and models for operating real-world system can benefit on both methods:

- **Model-based optimization can benefit on data-driven learning**: Training the DL forecaster with the awareness of the physical meaning of downstream tasks can become more economical and reliable.

- **Data-driven learning can benefit on model-based optimization**: Data-driven approaches can provide concrete uncertainty quantification and, in turn, improve the optimality and feasibility of model-based methods.

In recent years, it has drawn great attention within both operation research (OR) and deep learning communities regarding co-designing data-driven algorithms and model-based optimization techniques. Referring to Fig.1.1, it is further classified into two paradigms.

- **Sequential formulation on data-driven learning and model-based optimization**. In this framework, the data-driven and model-based components are still treated as separate two-stage procedures, but with some shared information between them. From the perspective of the model-based method, the outcome of a probabilistic forecasting model can be used to quantify the uncertainty set of stochastic or robust optimization. In physics-informed learning, the physics rules are embedded to reshape its parameter space [17]. In this setting, the objective of training the DL model is still to improve its accuracy.

- **Integrated Formulation on data-driven learning and model-based optimization (End-to-end learning)**. When the quality of the decisions is beyond the accuracy of the DL model, the system operator may want to explicitly train the DL model, in a way that can steer the model-based optimization to have more economical decision. In the paradigm of *contextual optimization*, it appears as early as in [18] and has recently shown a surge of interest with alternative names such as (task-aware) end-to-end learning, [19, 20], (smart) predict-and-optimize [21], integrated learning and optimization [16], as well as decision-focused learning [22, 23].

The training and inference procedure for sequential and integrated formulations are illustrated in Fig.1.3 and 1.4, respectively. Referring to Table 1.1, in some cases, the boundary between two formulations can be vague. For example, the sequential formulation can become integrated when the models are represented by an explicit function or when the solution of the down-stream optimizations can be analytically derived. Meanwhile, although both the integrated formulation (Fig.1.4) and the purely data-driven decision-making (Fig.1.2) try to minimize the task-aware objective for training the parametric model, the key difference is that in the purely data-driven approach, the downstream optimization is also modeled by trainable parameters, and the parametric space of forecast and decision is not separable [24]. Therefore, the integrated formulation is also more interpretable than the purely data-driven approach. Broadly speaking, integrated formulation can also be classified as physics-informed deep learning, since there is a physics model (the equilibrium condition represented by model-based optimization) encoded as a layer of neural network. However, this thesis treats it as a distinct subject because of its strong connection to optimization theorem and operational research.

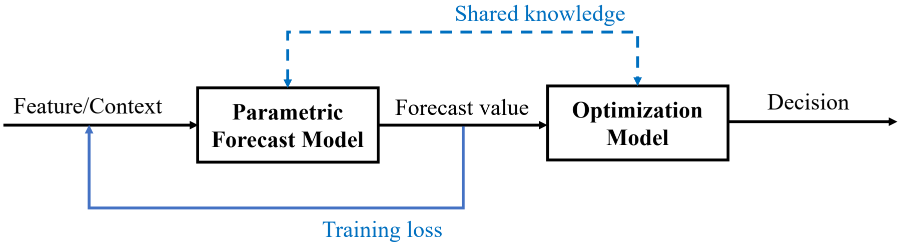


Figure 1.3: The inference and training (in blue) procedure of sequential formulation.

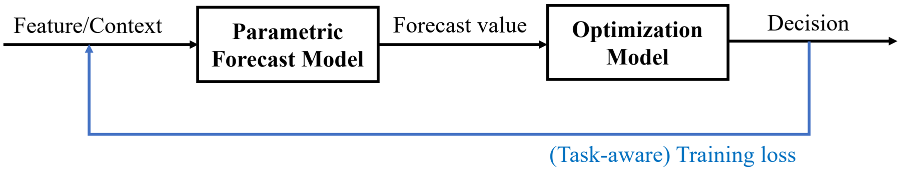


Figure 1.4: The inference and training (in blue) procedure of integrated formulation.

1.4 Introduction to Power System Operation and Cyber-Security

As discussed in the previous sections, this thesis mainly develops new sequential and integrated formulations of the data-driven and the model-based methods. Although the proposed learning frameworks are general, their performances are mainly verified in power system applications, such as power system operation and cyber security. Therefore, an overview of the two topics is presented in this section.

1.4.1 Power System Operation

This section provides a brief introduction to power system planning, operation, and control based on the textbook [25]. In general, these components are integrated into the economic layers of the power system, distinguished by various time dependence.

Spanning from one to twenty years, the planning problem includes the construction of new electric energy system or the reinforcement and expansion of an existing one. For example, transmission expansion planning is carried out with increased DERs, flexible load components, and the nexus of the multi-energy system [26]. Meanwhile, the cyber resiliency of the digitalized smart grid needs to be improved against unseen cyber intrusions [27].

Operations performed around a month prior to power delivery are named operation planning, which includes energy resource management and preventive maintenance of the facilities. In the day ahead, electrical power consumption and renewable energy generations are forecasted. Based on the forecast amount, the set-points of the controllable generating units are scheduled, and the reserves are prepared for the next-day operation. In the centralized system, the day-ahead operation solves a unit commitment (UC) problem to determine the on-off status and hourly set-points of the generator and reserves, with the purpose of minimizing the cost. In the market framework, the market clearing algorithm is solved to maximize social welfare. Preventive reserves are deployed one hour before delivery. Therefore, an economic dispatch (ED) problem is solved to ensure an economical, reliable, and secure power supply.

In a much higher resolution (e.g., several minutes), optimal power flow (OPF), security-constrained optimal power flow (SCOPF), and state estimation (SE) are solved to monitor and adjust system state. Instead of perusing maximum profit, the main criterion at this stage is to maximize security. In particular, the SE retrieves the system voltage magnitude and phase angles from the sensor measurements. Voltages can further inform the operation status and reveal any contingencies or anomalies in the grid. In SCOPF, active and reactive powers are adjusted by enforcing several contingency scenarios. Such ex-ante preventive actions ensure that the system can resolve the contingency by introducing appropriate corrections.

In real time, the generating units are dispatched to meet demand while meeting all the security constraints of the grid. Active power controllers use active power reserves to maintain the system frequency close to 60/50 Hz [28]. It also deviates the active power flows on tie-lines between different areas, based on the economic agreements. Meanwhile, reactive power reserves from the generating unit, the capacitor bank, the online tap changer (OLTC), and the power electronic inverter are controlled to maintain a healthy voltage profile [14].

This thesis focuses mainly on the operation stage problem, which is strongly related to centralized UC, ED, and SE, using the proposed sequential and integrated learning frameworks.

1.4.2 Cyber-Security in Power System

Overview

The advent of information technologies (ITs) has transformed the power grid into a complex cyber-physical power system (CPPS), introducing new risks due to two-way real-time communication among multiple parties [29]. However, this new opportunity also presents challenges in the safe and resilient operation of CPPS under cyberattacks [30]. Therefore, another key practical focus of the thesis is to improve the security and robustness of power system operations under attacks or uncertainties. The work of Musleh *et al.* [31] provides a comprehensive review of seven recent cyber attacks in the energy industry, identifying vulnerabilities in both physical and cyber layers. One notable example is the half-day grid blackout in December 2015 in Kiev, where more than 200,000 customers were affected. Subsequent investigations revealed that hackers had exploited vulnerabilities in the supervisory control and data acquisition (SCADA) protocol [32]. In general, cyber-physical attacks can be classified according to their unique target and delivery methodologies [31], such as denial-of-service (DoS) attacks in the network and communication layers where information flow packets are jammed or lost. In particular, recent research has predominantly focused on False Data Injection (FDI) attacks, exploring various dimensions such as DC microgrid operation [33, 34], energy markets [35], frequency regulation [36], and voltage regulation [37]. However, this thesis primarily addresses FDI attacks targeting state estimation.

False Data Injection Attacks

As described in Section 1.4.1, the control center retrieves the operational states of individual buses using the measurements observed from the remote terminal unit (RTU) and/or phasor measurement unit (PMU) at regular intervals, ranging from seconds to minutes. The estimated state plays an essential role in energy management systems (EMS) for tasks such as contingency analysis, automatic generation control (AGC), and load forecasting [38]. Consequently, the FDI attack is defined as the direct manipulation of measurements with the purpose of deviating from the estimated state, thus misleading the economic and stable operations of the EMS [39,40]. Recent advances in FDI attacks have exploited vulnerabilities in the Modbus/TCP protocol, circumventing detection by the bad data detector (BDD) in the control center [41–44].

Detecting FDI attacks efficiently without significantly affecting normal operation is essential for power system operation. Intuitively, employing a data-driven detector seems reasonable, given the stealthy nature of FDI attacks on model-based BDD. Meanwhile, a representative of the dynamic model-based approach, known as moving target defence (MTD), is recently introduced into the power system. MTD is designed to proactively alter the physical structure of the grid, enhancing its resilience against potential cyber threats.

Within the broader framework of blending data and model-based approaches, the detection of FDI attacks serves as a pertinent benchmark. It provides a tangible measure of the robustness and efficiency of the proposed framework, ensuring that the system remains resilient to cyber threats while maintaining normal operational functionality.

1.5 Literature Review

In this section, both data-driven and model-based detection methods for FDI attacks are reviewed. Furthermore, by assessing the strengths and weaknesses of these two methods, the motivation for the primary research focus on leveraging both data and model is articulated. Special attention is paid to the MTD, which has its own theoretical foundations. This section then moves to the implementation of blending data and model, and review on the literature on integrating load/renewable forecast with the centralized and market-based power system operations. It is essential to note that each subsection can be approached as a standalone monograph on relevant topic.

1.5.1 Detection on False Data Injection Attacks

The concept of stealth FDI attacks on DC state estimation is first verified in [41] where the attacker can bypass the model-based BDD with limited resources, even with protected remote terminal units (RTUs). Later, the FDI attack on AC state estimation is proposed in [42] along with vulnerability analysis. In general, three hierarchical approaches can be implemented to deal with the FDI attacks, namely protection, detection, and mitigation [44]. Protection is an attack prevention mechanism that can reject common attack attempts. However, it is too hard and costly to protect all measurement units in the grid, and a complete rejection of the attack is unrealistic [45]. The second stage is attack detection, where model-based and data-driven detectors are two prevalent methods [31]. The third approach is attack mitigation. For example, in [46], a trilevel optimization algorithm is proposed to restore the operation after attack with minimum restoration duration.

Model-Based Detection Method

Traditional BDD involves solving the static state estimation problem and calculating the deviation between true and reconstructed measurements [38]. However, since the static model cannot capture the dynamics of the power system, it is not effective in detecting structured attacks [47]. As a result, dynamic state estimation is applied to capture temporal correlations in load and generation patterns and alerts the system operator when this trend is violated. The dynamic state estimation based on the Kalman filter is one of this kind [48]. However, the model-based detectors fully use the knowledge of the system model and dynamics, which can be easily interpreted and adopted by the system operator. Although the static model is reliable for decision making, it can be easily targeted by reconnaissance attacks. The grid topology and parameters can be retrieved by deliberate attackers using topology identification algorithms. Once the grid knowledge is learned, the attacker can formulate the attack vector to bypass the model-based detector.

Data-Driven Detection Method

As an improvement of the number and resolution of the grid measurements, the data-driven method is armed to model complex grid dynamics and uncertainties. In general, learning algorithms for detecting attacks can be classified into *supervised* and *unsupervised* learning. In the supervised setting, detection is achieved directly by learning a mapping from the input (feature) space $\boldsymbol{x}_i \in \mathcal{X}$ to the binary classification $y_i \in \mathcal{Y} = \{0, 1\}$, for example, $f_\theta : \mathcal{X} \longrightarrow \mathcal{Y}$. The model fitted on the training data-set can be directly used to classify the legitimacy of the test set. Support Vector Machine (SVM) [49], Naive Bayesian Classifier (NBC) [50], and Decision Trees (DT) [51] belong to this category.

To enrich a balanced dataset for supervised learning, attack samples are synthesized in the literature [52]. However, the synthetic attack data may not be representative for the actual attack attempts, degrading the detection performance of the supervised classifier. To overcome the problem, unsupervised and semi-supervised methods are considered to detect the FDI attacks by learning the latent representation of the legit measurements. Let $\mathcal{X}$ be the input (feature) space and $\mathcal{Z}$ be the latent space. The unsupervised learning can be represented by $f_\theta : \mathcal{X} \longrightarrow \mathcal{Z}$ where the latent representation can be used for clustering or dimension reduction. Unlike the supervised detector, an implicit classifier should be built on the latent space $\mathcal{Z}$ to detect the attack. The un/semi-supervised learning approach includes isolation forest [53], semi-supervised support vector machine [54], autoencoder [55], and prediction-based algorithm where a predictor is built on normal data and the attack is detected by violating the distribution of the prediction errors [56]. Despite the high detection rate of unsupervised detectors, they suffer from high FPR on

Table 1.2: Comparison between data-driven and model-based FDI attack detection.

	Data-driven Detector	**Model-based Detector**
Method	Use data to model the dynamic of the system	Use the existing system model
Pros	(1) Fast real-time response (2) No need for model information	(1) High interpretability (2) Controllable FPR
Cons	(1) Less interpretability (2) Uncontrollable FPR (3) Vulnerable to adversarial attacks (4) Inefficient and unbalanced data	(1) Targeted by intelligent attacker (2) Model uncertainty and mismatch (3) Time-consumed real-time optimization (4) Interfere normal operation

legit measurement under the test set and roundabout training target during training. For example, in [57], up to 20% FPR is committed to achieve 90% TPR. As data availability is of high priority in CPSG, continual false alarms from a data-driven detector cause frequent contingencies and overload response resources, compromising the operator's confidence in the detector.

Table 1.2 compares the detections of FDI attacks based on model and data. Data-driven and model-based detectors can be compensated in various perspectives. For example, uncertainties in model-based detector can be mitigated by data-driven calibration [58]; the computational burden of real-time operation can be solved by event-triggering by the data-driven detector [59]. In contrast, the interpretability of data-driven detector can be improved by embedding physical information from the system model [60], such as the power flow equations. Based on the above comparison, detection on FDI attacks becomes a natural benchmark to test the effectiveness of blending data and models.

1.5.2 Moving Target Defence

As the power system operates quasi-statically, intruders have enough time to learn the system parameters and prepare FDI attacks [61–63]. As a result, it is viable to invalidate the attacker's knowledge by proactively changing the system configuration. Moving target defence, which is conceptualized first for information technology (IT) security, utilizes this proactive defence idea [64]. With D-FACTS devices, the control center can alter the reactances of the transmission lines to physically change the system parameters that are unknown to the attackers, as shown in Fig. 1.5. Unlike MTD in IT system, the physical structure of the power system and the attack surface should be explicitly considered when

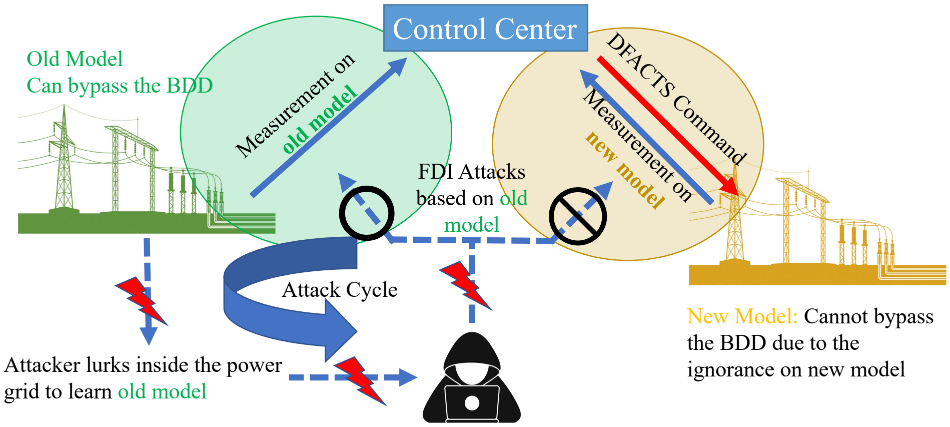


Figure 1.5: The working flow of MTD.

operated in the cyber-physical smart grid.

There are three main problems for the design of MTD, namely *'what to move'*, *'how to move'*, and *'when to move'*. In detail, 'what to move' finds the optimal placement of D-FACTS devices at the planning stage so that the attack surface is minimized; 'How to move' determines the set-points of the D-FACTS devices during the operation to maximize the detection rate while reducing the extra operational cost. Finally, 'when to move' determines the occasion to send the MTD command to the field devices, either *periodically* or *event-triggered*.

To this end, a thorough review on the MTD algorithms is summarized in Table 1.3 with regard to their different perspectives on 'what to move', 'how to move', and 'when to move'. Note that in Table 1.3, only relevant researches to this thesis are summarized, that is, the MTD to detect the FDI attack against SE. To better analyze the ideas, four stages of the development of MTD can be identified.

- **Stage One.** Initially, MTD research involves using random placement and reactance perturbations to expose FDI attacks [65–67]. However, it has been shown that the so-called "naive" applications cannot guarantee an effective detection on stealthy FDI attacks.

- **Stage Two**. At this stage, the 'what to move' is solved, e.g. where to locate the D-

FACTS devices to maximize the detection performance. The authors in [68] and [69] demonstrate that the effectiveness of MTD depends on the rank of the composite pre- and post- MTD measurement matrices. Furthermore, Liu, *et al.* [70] and Zhang *et al* [71] investigate the D-FACTS devices placement in the planning stage to maximize effectiveness while minimizing investment budget. Note that in this stage, the FDI attack is only formulated based on DC SE. Meanwhile, it is assumed that there is no measurement noise. Therefore, the guaranteed detection performance may not work in practice.

- **Stage Three**. At this stage, more realistic operation condition and more intelligent attackers are considered to improve the MTD performance. For example, AC power flow model and measurement noise are included. In the meantime, MTD operational cost, hiddenness, and event-triggereing are investigated. That is, they are trying to solve the 'how to move' and 'when to move' problem. In detail, early-stage MTD researches only consider the DC power flow model. However, many researches have shown that defence based on DC models may not result in desirable performance on AC-based attack. Therefore, explicit modeling of AC attack and defence are discussed in [72], though a certain linearization is required. The authors of [73] analyze the effectiveness of MTD in real-time using the minimal principal angle metric and numerically show the relationship between the angle and the average detection rate. As cyber attack is rare in real-time power grid operation, the cost of frequent changes on grid parameters can hardly be accepted by the system operator. Therefore, [73] proposes to combine the detection performance of MTD with the OPF problem to simultaneously minimize the generator cost. Liu, *et.al.* [72] explicitly increases the effectiveness in the cost function. Xu, *et.al.* [74] derives a robust metric to guarantee the effectiveness of MTD on unknown attacks. Recently, *hidden* MTD is proposed to compete with vigilant attackers who can perform SE and BDD to verify the integrity of the grid parameters [72, 75–77]. This is actually a natural point to consider, as the attacker has to do state estimation to formulate the attack.

Moreover, Higgins *et.al.* [78] suggests perturbing the reactance through Gaussian watermarking to prevent the attacker from inferring the new system parameters. Regarding on 'when to move' problem, in most of the literature, MTD is synchronized with SE and OPF to detect the most frequent attack, while event-triggered approach is limitedly analyzed in [59, 79]. The inevitable cost on MTD is still significant when considering the small chance that the grid is targeted by FDI attacks. The question of how to balance detection performance and extra operational cost remains an open question.

- **Stage Four**. Since MTD cannot detect all FDI attack due to the restriction of grid topology, recent works have combined MTD with other defence techniques, such as cyclic MTD [71] where the entire attack space is covered by multiple and successive MTD perturbations. In addition, the unprotected subspace can also be protected by meter protection [76,80]. The idea of MTD is also extended to encoding encryption in [81]. Meanwhile, the applications of MTD beyond the static state estimation are researched. For example, [82] applies MTD to detect a Stuxnet-like attack in dynamic system. Authors in [83–86] apply the MTD on converter in microgrid and unbalanced distribution network with the voltage stability constraint being considered. In addition, the MTD is also used for vulnerability assessment metric in [87].

Table 1.3: Literature Review on MTD (sorted by the first appeared publication years).

Ref.	**Year**	**Description**
What to Move		
[65–67]	2012-14	Randomly install and perturb D-FACTS devices in all branches.
[68, 88]	2017-18	Link detection to the rank of the composite matrix.
[89, 90]	2019-21	D-FACTS placement for cyber-physical attack.
[75]	2019	Prove the sufficient condition for hidden MTD under DC condition and the contradiction between hidden and complete MTD.
[91]	2019	Give the systematic condition for complete and incomplete MTD.
[70]	2020	Give algorithm to place D-FACTS devices for complete and incomplete MTD.
[92]	2020	Enhance MTD with meter protection that can recover the compromised state.
[93]	2020	Give the systematic condition to complete and incomplete MTD. Demonstrate that the perturbation almost cannot change the rank condition.
[76]	2020	Prove the attacker's sufficiency to detect MTD. Co-design MTD and meter protection to achieve hidden MTD.
[73]	2021	Propose that the detection performance is proportional to the minimum principal angle between the subspaces of the pre- and post-MTD matrices.
[77]	2021	Propose a sufficient condition for hidden MTD. Placement algorithm for hiddenness while maximizing the rank of the composite matrix.
[94]	2021	Propose detection upper bound based on the rank of the composite matrix and placement algorithm to improve the detection.
[72]	2022	Derive the residual-based criterion for MTD hiddenness and effectiveness by linearization on the power flow equations.
[71]	2022	Propose repeatedly covering different subspaces in incomplete MTD.
How to Move		
[88]	2017	Iterative perturb based on the minimum generator cost.
[95]	2018	Minimize difference between pre- and post-MTD power flows and loss.
[68]	2018	Iterative perturbation based on the loss-to-reactance sensitivity matrix.
[89, 90]	2019	Balance the generator cost and MTD effectiveness.
[75]	2019	Analyze that the hidden MTD can also reduce the extra generator cost.
[70, 93]	2020	Minimize the generator cost while perturbing a minimum amount.
[78]	2021	Add physical watermarking on the branch parameter for hiddenness.
[73]	2021	Minimize generator cost while fulfilling detection as hard constraint.
[77]	2021	Minimize generator cost and difference between pre- and post-MTD power flows. Maximize the reactance perturbation to enhance the detection performance.
[72]	2022	The effectiveness and hiddenness of MTD are balanced by optimization problem.
[96]	2022	Reduce the difference between post- and pre-MTD generator costs while perturbing a minimum ratio of the D-FACTS devices.
[74]	2022	Prove the connection between detection and the principal angle. Propose an algorithm for complete and incomplete MTD.
When to Move		
[59]	2021	The MTD is triggered by data-driven detector but randomly perturbed.
[79]	2023	The hidden and complete MTD is triggered and informed by data-driven detector.

1.5.3 Algorithms on Blending Data-driven and Model-based Approaches

This section draws on relevant researches for decision making with data, from both theoretic and practical perspectives. Specifically, it focuses on the co-design of data and model, especially with the integrated formulation, i.e. the end-to-end learning.

In contextual optimization, it is natural to use DL model to forecast the uncertain parameters in constraint optimization. In a sequential formation, the forecast error of the trained forecaster can be used to construct the uncertain set of robust optimization [79] , the conditional distribution of stochastic optimization [97], or the ambiguity set of distributional robust optimization [98].

As shown in Table 1.1 and Fig.1.4, the integrated formulation has three sequentially connected components: (a). A parametric forecast model that maps the contextual information (e.g. the input feature) to the interest of forecast; (b). Optimization models that take the forecast as input and return decisions; and (c). A task-aware loss function that encodes the ultimate goal of decision making in the whole system. The key of this framework is to infer the cost of down-stream task(s) directly into the training objective. Therefore, depending on the different approximations and solution algorithms in the optimizations, the solution of the integrated formulation can be broadly divided into direct solution on optimization, unrolling the optimization, differentiable layer, and surrogate model.

In general, integrated formation on data and model is essentially an optimization programming problem. For instance, many E2E learning can be mathematically formulated as bilevel optimizations in which the upper level is the expected task-aware cost on the training dataset and sub-level problems are represented by down-stream optimizations. Therefore, a direct solution is applicable if the E2E formulation follows a simple pattern. For instance, if the forecast model is linear and down-stream tasks are linear or quadratic, the bilevel optimization can be converted into mixed integer linear programming (MILP) [99], in which exact solution can be found. However, when the down-stream optimizations are non-convex, exact solution may not be guaranteed and iterative algorithm is used [100]. Meanwhile, when NN is used as forecast model, direct invoking optimization software may not be a viable option.

When the gradient descent method is used to train the E2E model, the gradient from the task-aware cost to the forecast value is required. From the optimization point of view, since the task-aware cost can be analytically written as a function of the optimal decision of down-stream optimizations, it is necessary to compute the exact or approximate Jacobian of the optimal decision to the parameter of the optimization problems. By constructing the differentiable optimizer, automatic differentiation packages are able to construct the computation graph for the back-propagation [101]. Therefore, in the unrolling method, the iterative solution process is encoded as NN layers [102]. The computation graph is stored in the forward pass of NN and the gradient of each iteration exists. Since each iteration can be viewed as a fixed operation on the previous result, a recurrent neural network (RNN) can be used to reduce memory burden [103].

Unlike the unrolling approach, the differentiable layer method does not need to record the intermediate iteration in the computation graph. Instead, the down-stream optimizations are solved by off-the-shelf solvers in the forward pass. An implicit function is built to link the parameter and the optimal solution of the forecast parameter. Under some minor assumptions such as non-singularity, the Jacobian matrix exists according to the implicit function theorem [104]. Practically, OptNet [105] applies the implicit function theorem to denote the Jacobian after formulating the Karush-Kuhn-Ticher (KKT) condition of parametric quadratic programming (QP) which is further extended to disciplined convex programming in CvxpyLayers [106]. However, in some real-time problems, the Jacobian between the optimal decision and forecast parameter is not well defined. Therefore, surrogate models of the original optimization can be built to enforce the desired continuity, differentiability, and convexity, as an alternative to the KKT condition. For example, although linear programming (LP) is a special type of QP, OptNet is not applicable, as the gradient is zero or undefined everywhere. To address the problem, a quadratic penalization is added in the objective so that the LP is second-order differentiable [22]. To find an equivalent continuous optimization, the cutting planing algorithm is introduced, and the resultant linear programming is equivalent to the original mixed integer linear programming (MILP) at the optimal solution [107].

1.5.4 Integrating Energy Forecasting and Power System Operation

E2E learning has also drawn great attention for economic power system operations in recent years. As discussed in Section 1.4.1, various decision-making tasks span different time horizons with forecast elements included. This includes load and renewable forecasting at the day-ahead level, generator dispatch at both day-ahead and intra-day levels, state estimation, contingency analysis at higher resolution, and real-time safety control, among others [25]. The inherently anisotropic nature of power systems requires a specifically tailored DL framework to effectively address the unique challenges posed by these decision-making tasks.

The main argument of introducing integration on the data and model is that the operating cost of the power system is highly asymmetric and depends on the varying operating point. For example, the lack of generation is penalized more than the over-generation. Therefore, training the forecaster with a symmetric accuracy-driven objective, such as MSE loss, is not aligned with minimizing the operating cost. In practice, some system operators notice that the forecast load can influence the profit of down-stream operations. For example, the California Independent System Operator manually modifies the forecast load to increase ramping capacity [99]. In addition, the relationship of operating cost and forecast error is non-linear due to the existence of physical constraints. Based on the methodologies introduced in the previous section, the recent literature is summarized in Table 1.4. Similarly, the key to different methods is to find a way to connect the forecast target to the optimality of the down-stream optimization(s). In addition, the type of parametric forecast model and the type of down-stream optimizations are also highlighted.

In general, load and renewable are prevalent as the forecast targets. Recently, inertia forecasting has also been discussed due to the high penetration of renewables. As for the down-stream tasks, centralized operation (including dispatch and re-dispatch problems) and energy market (including both day-ahead and real-time clearing) are mostly discussed.

Table 1.4: Literature Review on E2E learning in power system.

Ref.	Year	Method	Forecaster	Decision Model
[19]	2017	Differentiable layer	Load distribution (MLP)	Dispatch problem (Stochastic QP)
[108]	2018	Surrogate model	Wind generation (Boosted regression tree)	Wind regulation (Analytic)
[109]	2019	Surrogate model	PV Generation (Analog ensemble), spot price (SVR), regulation price (KNN)	Day-ahead and real-time market (Analytic)
[110]	2020	Direct solution (Single level)	Renewable energy bid (Linear)	Day-ahead and real-time market (LP)
[111]	2021	Differentiable layer	Load distribution (MLP)	Economic dispatch (Stochastic QP)
[112]	2021	Direct solution (Single level)	Load and Reserve (Linear)	Dispatch and re-dispatch problem (LP)
[113]	2021	Direct solution (Single level)	Wind generation interval (Linear)	Reserve provision (MILP)
[114]	2022	Direct solution (Single level)	Renewable energy (Linear)	Dispatch (MILP)
[115]	2022	Direct solution (Bilevel)	Wind generation interval (Linear)	Wind regulation (Robust LP)
[116]	2022	Surrogate model	Load (Linear & MLP)	Dispatch and re-dispatch problem (QP)
[117]	2022	Surrogate Model	Electricity price (ResNet)	Energy storage arbitrage (MILP)
[100]	2022	Direct solution (Bilevel)	Renewable and reserve (Linear)	Dispatch and re-dispatch (MILP)
[118]	2022	Differentiable layer	Load (LSTM)	Dispatch problem (Stochastic LP)
[119]	2022	Surrogate model	Renewable (Decision tree)	Day-ahead market (Stochastic QP)
[99]	2023	Direct solution (Bilevel)	Load (Linear)	Dispatch and re-dispatch problem (Linear)
[120]	2023	Direct solution (Single level)	Inertia (Linear)	Day-ahead and real-time market (QP)
[121]	2023	Surrogate model & Differentiable layer	Load (LSTM)	Dispatch and re-dispatch problem (MILP)
[122]	2023	Differentiable layer	Wind generation (GRU)	Dispatch problem (Linear)
[123]	2023	Differentiable layer	Renewable energy (MLP)	Dispatch and re-dispatch problem (Linear)
[124]	2023	Surrogate model	Load (MLP)	Dispatch and re-dispatch problem (MILP)
[125]	2023	Reinforcement learning	Wind generation interval (MLP)	Dispatch and re-dispatch problem (Robust QP)
[126]	2023	Differentiable layer	Load (Linear & CNN & MLP-Mixer)	Dispatch and re-dispatch problem (QP)
[20]	2023	Differentiable layer	Load (MLP)	Dispatch and re-dispatch problem (LP)

The direct solution method, either formulated as single-level or bilevel optimization, is more commonly used in power system operation than other fields. As mentioned above, only linear forecasters are used in the direct solution method, as NN typically requires using the stochastic gradient method (SGD) to train. To overcome the limited complexity of linear model, an extreme learning machine is used to train only the last linear layer of NN [113,115] while enjoying its high representation ability. The differentiable layer method has also been implemented, as most of the operation of the power system operation can be modeled or simplified as QP or LP [20, 122, 123, 126].

In the surrogate model method, by noticing the asymmetric pattern in operating cost, [108, 116] learn a piece-wise linear map from the forecast error to the cost deviation, which is used as the NN training loss. In addition, most of the application of surrogate model is to approximate the non-differentiable task-aware cost to enable back-propagation. For example, [121] applies a similar idea in [107] to find an equivalent LP of MILP through branch and bound. The resultant LP can then be treated as a differentiable layer. In some applications, the surrogate model is derived for upper bounding the non-differentiable task-aware cost [117] so that the worst performance of E2E learning can be guaranteed. The energy market without constraints can also be solved by the surrogate method, as the only asymmetry is introduced by the unbalanced cost between the energy sold by the producer and the actual energy delivered [109].

Recall that E2E learning comes from contextual optimization to quantify uncertain parameters. Many power system operations also try to combine probabilistic forecast with stochastic optimization. It is highlighted that such an implementation not only provides less conservative uncertainty set to the decision making, the uncertainties of the forecaster can also be reshaped to be more “certain”. For example, the distribution of the forecast is shifted toward the direction with high profit. Two distinct formulations have been reported. First, when the down-stream task is stochastic programming (SP), the mean and variance of the forecast can be predicted. Assuming that the forecast error follows a normal distribution, sequential quadratic programming is used to find the optimal decision, and the differentiable layer method can be applied for convex SP [19, 111, 118]. Second, the prediction interval is an effective tool for quantifying uncertainty and usually serves as an input to down-stream robust optimization [113, 115, 125].

Furthermore, the training objective is divided into two categories. In most of the literature, regret has been used as the training objective of E2E learning, which measures the difference between the optimal objective based on the ground-truth deterministic parameter and the objective parameterized by the forecast value. Since the optimal objective is constant with respect to the DL model, it is equivalent to the task-aware cost. Optimal decision/objective imitation has been adopted in [114, 122] in which the MSE loss between the optimal objective and the objective parameterized by the forecast, or between the optimal decision and the decision parameterized by forecast, is used as the training loss. In addition, since [114, 122] only consider single-stage dispatch problem, regret loss cannot be used, as the loss is not bounded by ground-truth load or renewable. Consequently, many studies argue that regret is more intrinsic than MSE loss and the two-stage problem should be considered for power system applications [20, 100]. It is noted that even for purely model-based optimization without forecaster, the bilevel formulation can mimic real-time implementation more reasonably by respecting the sequence of decision-makings [127, 128]. For the non-parametric forecaster, such as those modeled by decision trees in [119], the task-aware cost can be used to replace the splitting criterion.

In addition to using end-to-end models to train the DL forecaster, the task-aware objective can be used to evaluate the influences or values of the input samples of the trained forecaster. In [121], the authors evaluate the value of the dataset from different sectors in a multi-energy system by integrating two-stage optimizations. In machine unlearning [126], the task-aware cost is used to assess the influence of consumer data on the test set performance. A trade-off between unlearning completeness and whole system operation cost can be built. The authors in [109] also show that simultaneous forecasting on more than one target organized by task-aware loss can result in more revenues, as the individual forecasting models can adapt to each other's forecast errors. The authors in [120] also include risk aversion in the optimization model.

1.6 Contributions

This thesis revolves around a central research theme, namely, examining the advantages of integrating data and model, particularly in the context of power system operation under uncertainties and attacks. The principal contributions of this research can be succinctly summarized as follows.

- **Intelligent decision-making with combined data and model-based algorithms.** In general, the advantages of both sequential and integrated formulations in combining data and model for power system operations are explored, in comparison to purely data-driven and model-based approaches.

- **Robust MTD.** The definition of MTD effectiveness has been broadened, moving beyond the rank between the composite pre- and post-measurement matrix to consider the principal angles between their corresponding subspaces. This extension allows the system operator evaluate their MTD strategy under certain level of measurement noise and assumption on attack strength. A robust MTD algorithm is proposed, in the presence of various system topologies and the locations of D-FACTS devices.

- **Sequentially designed data-driven detector and robust MTD.** A sequential detector is comprehensively designed by encoding both data-driven detector and robust MTD. A long short-term memory autoencoder (LSTM-AE) network is built to detect FDI attacks and the approximate set of the corresponding normal measurements are recovered in a physics-informed way. A new robust MTD algorithm is formulated as bilevel optimization to improve the hiddenness of MTD while satisfying the a minimum detection rate. The shared information between the model and the data contributes to reducing the FPR of the data-driven detector, thereby significantly improving its reliability. The extra operational cost caused by the conservative nature of the robust MTD becomes more economical due to the event-triggering mechanism of the data-driven detector and the informative data-driven attack uncertainty set.

- **Uncertainties in E2E learning and unified adversarial training.** The investigation delves into understanding how uncertainties in the unpredictable parameters of optimization can impact the generalization capabilities of the trained forecaster. A precise, task-aware adversarial attack is formulated, targeting the input of the E2E model with a piece-wise linear forecaster and QP decision-making. Drawing inspiration from adversarial training, a unified framework is designed to address uncertainties in the input features of DL and unpredictable parameters in optimization.

- **Task-aware machine unlearning.** A novel task-aware machine unlearning algorithm is designed to eliminate the impact of specific segments of the training dataset on the trained load forecast model. This algorithm seamlessly integrates into the E2E learning framework, evaluating the contribution of unlearned samples based on their impact on operational cost. Additionally, it introduces a task-aware sample re-weighting algorithm aimed at striking a balance between the completeness of machine unlearning and the performance of the forecasting model.

1.7 Thesis Outline

This thesis is divided into two parts, comprising a total of four technical chapters. An overview of the thesis can be found in Fig.1.6.

Part I centers on the integration of deep learning and power system optimization problems, utilizing two distinct but sequential steps, namely the sequential formulation for co-designing data and model algorithms. To illustrate the effectiveness, the framework is applied to detect the FDI attacks and the implementation of MTD. Despite the separate operation of deep learning and model-based approaches, it is shown that the deep learning algorithm can mitigate the conservatism inherent in the model-based approach, especially under robust settings. Additionally, it also explores how incorporating physics information from the power system can enhance the feasibility of the deep learning. This part encompasses two chapters:

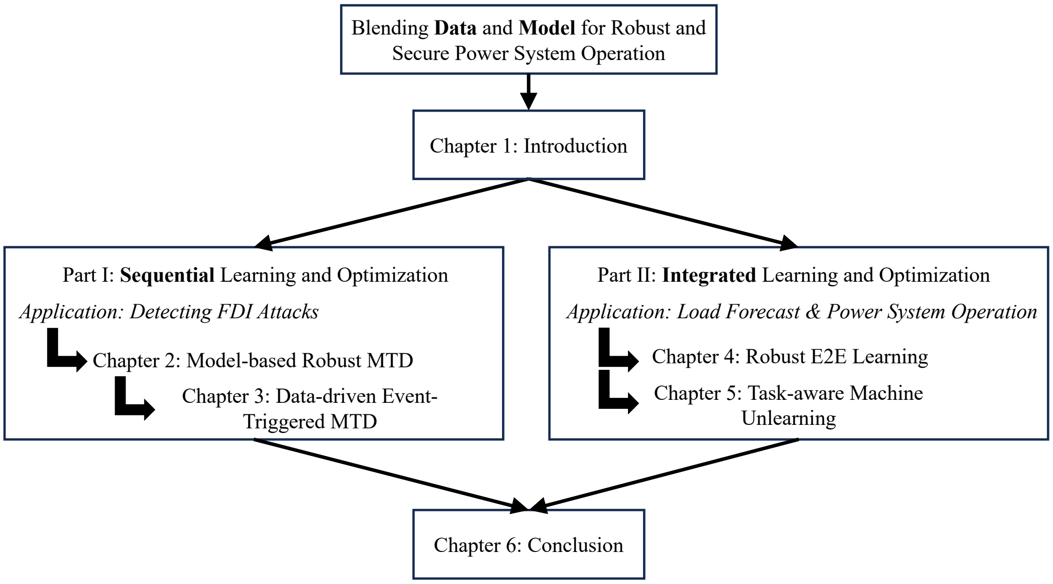

Blending **Data** and **Model** for Robust and Secure Power System Operation
Chapter 1: Introduction
Part I: **Sequential** Learning and Optimization
Application: Detecting FDI Attacks
Chapter 2: Model-based Robust MTD
Chapter 3: Data-driven Event-Triggered MTD
Part II: **Integrated** Learning and Optimization
Application: Load Forecast & Power System Operation
Chapter 4: Robust E2E Learning
Chapter 5: Task-aware Machine Unlearning
Chapter 6: Conclusion

Figure 1.6: Outline of the thesis.

- Chapter 2 investigates robust MTD for detecting FDI attacks on power system state estimation. In particular, this chapter proposes three theoretical principles related to principal angles between the pre- and post- MTD subspaces of the system to maximize the detection probability under noisy measurements. The proposed robust MTD algorithm will be further used as the model-based optimization in the sequential data-model pipeline of detecting FDI attacks in the following chapter.

- Chapter 3 designs a physical-informed deep learning-based FDI attack detector and identifier to trigger the robust MTD with hiddenness awareness. This sequential data-model detection algorithm can effectively reject the false alarms from the data-driven detector and reduce the size of the uncertainty set of the robust optimization. Consequently, the accuracy and reliability of the data-driven detector can be improved and the operational cost of MTD can be reduced.

Part II of the thesis delves into the study of an integrated forecasting and decision-making framework, conceptualized as an end-to-end training algorithm for DL. The emphasis is on exploring the robustness of this learning framework and extending its application to machine unlearning. This section comprises two chapters:

- Chapter 4 delves into the exploration of uncertainties in E2E learning in a unified manner. The chapter systematically addresses uncertainties in both the input of the DL forecaster and the unpredictable parameters of constrained optimization. To mitigate these uncertainties, an algorithm based on adversarial training is proposed. The theoretical framework is validated through applications to power system operation problems.

- Chapter 5 introduces the concept of machine unlearning to the load forecasting problem and explores the misalignment between accuracy-driven and objective-driven machine unlearning. A task-aware machine unlearning approach is proposed to strike a balance between the completeness of unlearning and the performance of the model.

In each chapter, supplementary yet pertinent background information and literature reviews are presented. Therefore, each chapter is self-contained and can be read as a monograph for a relevant topic. The final chapter of the thesis encapsulates the key findings and proposes directions for future work in each discussed topic. Mathematical proofs and additional experiment results are available in the Appendix.

1.8 Publications

This thesis is partially based on the following peer-reviewed journals and conferences:

1. **W. Xu**, I. M. Jaimoukha and F. Teng, "Robust Moving Target Defence Against False Data Injection Attacks in Power Grids," in *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 29-40, 2023, doi: 10.1109/TIFS.2022.3210864.

2. **W. Xu**, M. Higgins, J. Wang, I. M. Jaimoukha and F. Teng, "Blending Data and Physics Against False Data Injection Attack: An Event-Triggered Moving Target Defence Approach," in *IEEE Transactions on Smart Grid*, vol. 14, no. 4, pp. 3176-3188, July 2023, doi: 10.1109/TSG.2022.3231728.

3. **W. Xu**, F. Teng, "Task-Aware Machine Unlearning and Its Application in Load Forecasting", in *IEEE Transactions on Power Systems*, early access, 2024, doi: 10.1109/TPWRS.2024.3376828.

4. **W. Xu**, J. Wang, and F. Teng, "E2E-AT: A Unified Framework for Tackling Uncertainty in Task-aware End-to-end Learning", in *The proceedings of 38th Annual AAAI Conference on Artificial Intelligence (AAAI-24)*, vol. 38, no. 14, pp. 16220-16227, Mar. 2024.

5. **W. Xu**, I. M. Jaimoukha and F. Teng, "Physical Verification of Data-Driven Cyberattack Detector in Power System: An MTD Approach," *2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Novi Sad, Serbia, 2022, pp. 1-5, doi: 10.1109/ISGT-Europe54678.2022.9960436.

6. **W. Xu** and F. Teng,"Availability Adversarial Attack and Countermeasures for Deep Learning-based Load Forecasting," *2023 IEEE Belgrade PowerTech*, Belgrade, Serbia, 2023, pp. 01-06, doi: 10.1109/PowerTech55446.2023.10202786.

7. F. Bellizio, **W. Xu**, D. Qiu, Y. Ye, D. Papadaskalopoulos, J. Cremer, F. Teng, and G, Strbac,"Transition to Digitalized Paradigms for Security Control and Decentralized Electricity Market," in *Proceedings of the IEEE*, vol. 111, no. 7, pp. 744-761, July 2023, doi: 10.1109/JPROC.2022.3161053.

Publications that are not included in this thesis include:

1. J. Wang, **W. Xu** (co-first author), Y. Gu, W. Song, T. Green, "Multi-agent reinforcement learning for active voltage control on power distribution networks," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 3271-3284, 2021.

2. M. Higgins, **W. Xu**, F. Teng, T. Parisini, "Cyber–physical risk assessment for false data injection attacks considering moving target defences". in *International Journal of Information Security*, vol. 22, pp. 579–589, 2023, doi: 10.1007/s10207-022-00621-7.

Part I

Sequential Learning and Optimization for Detecting False Data Injection Attacks

Chapter 2

Robust Moving Target Defence Against False Data Injection Attacks in Power Grids

This chapter focuses on the model-based part in the framework of “sequential learning and optimization”, on which a data-driven detector will be added in the next chapter. Converting from the noiseless assumptions in the literature, this chapter particularly investigates the MTD design problem in a noisy environment and proposes the concept of robust MTD to guarantee the worst-case real-time detection rate against all unknown FDI attacks. The remainder of this chapter is organized as follows. The preliminaries are summarized in Section 2.1; Analysis on MTD effectiveness is presented in Section 2.2; Problem formulation and proposed robust algorithms are presented in Section 2.3; Case studies are given in Section 2.4 with conclusions in Section 2.5.

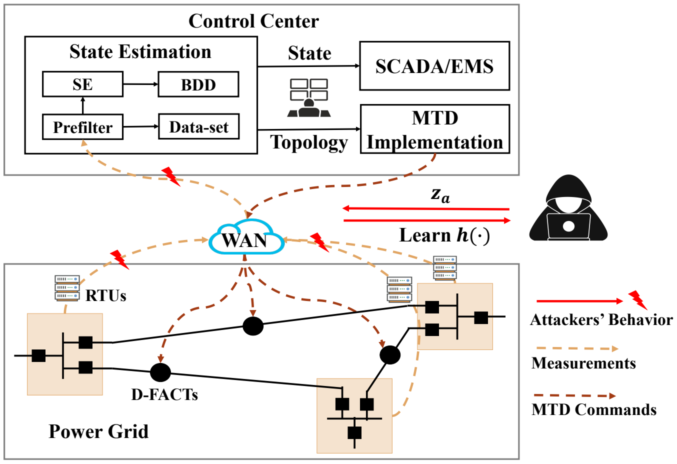


Figure 2.1: EMS with injection attacks and MTD in CPPS.

2.1 Preliminaries

2.1.1 Notations

In this chapter, vectors and matrices are represented by bold lowercase and uppercase letters, respectively. The p-norm of $\boldsymbol{a}$ is written as $\|\boldsymbol{a}\|_p$. The column space of $\boldsymbol{A}$ is $\mathcal{A} = \text{Col}(\boldsymbol{A})$. The kernel of a matrix $\boldsymbol{A}$ is represented as $\text{Ker}(\boldsymbol{A})$. The rank operator is written as $\text{rank}(\boldsymbol{A})$. $\boldsymbol{P_A} = \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$ represents the orthogonal projector to $\text{Col}(\boldsymbol{A})$ while $\boldsymbol{S_A} = \boldsymbol{I} - \boldsymbol{P_A}$ represents the orthogonal projector to $\text{Ker}(\boldsymbol{A}^T)$. The set of singular values is $\sigma(\boldsymbol{A}) = \{\sigma_1(\boldsymbol{A}), \sigma_2(\boldsymbol{A}), \ldots, \sigma_{\min\{m,n\}}(\boldsymbol{A})\}$. The spectral norm is $\|\boldsymbol{A}\|_2 = \max_i \sigma_i(\boldsymbol{A})$ and the Frobenius norm is $\|\boldsymbol{A}\|_F$. We use the symbol $(\cdot)'$ to indicate the quantities after MTD and $(\cdot)_a$ to indicate the quantities after the attack. The matrix operator $\circ$ represents the Hadamard product. Other symbols and operators are defined in the chapter whenever appropriate.

2.1.2 System Model and State Estimation

The power system can be modeled as a graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ with $|\mathcal{N}| = n + 1$ number of buses and $|\mathcal{E}| = m$ number of branches. For each bus, its complex voltage is denoted as $\boldsymbol{\nu} = \boldsymbol{v}\angle\boldsymbol{\theta}$; and for each branch, the admittance is denoted as $\boldsymbol{y} = \boldsymbol{g} + j\boldsymbol{b}$. The power balances can be modeled by a set of nonlinear equations $\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{\nu}) + \boldsymbol{e}$ where $\boldsymbol{z} \in \mathbb{R}^p$ is the sensor measurement and p is the total number of sensor measurements; $\boldsymbol{h}(\cdot) \in \mathbb{R}^p$ is the power balancing equation; $\boldsymbol{\nu} \in \mathbb{R}^{2n+1}$ is the system state consisting of voltage magnitudes at all buses and phase angles at non-reference buses. The measurement noise vector $\boldsymbol{e}$ follows a zero-mean independent Gaussian distribution with diagonal covariance matrix $\boldsymbol{R} = \text{diag}([\sigma_1^2, \sigma_2^2, \cdots, \sigma_p^2])$.

As shown in Fig. 2.1, the control center is equipped with state estimation (SE) which serves as a bridge between remote terminal units (RTU) and the energy management system (EMS) [129]. The goal of SE is to retrieve the system states at all buses from the noisy sensor measurement $\boldsymbol{z}$. In particular, $\boldsymbol{z}$ is composed with active and reactive power injections and flows such that the system equations $\boldsymbol{h}(\cdot)$ is represented by power flow equations [129]:

$$P_i = v_i \sum_{j=1}^{n} v_j \left(g_{ij} \cos\theta_{ij} + b_{ij} \sin\theta_{ij} \right) \tag{2.1a}$$

$$Q_i = v_i \sum_{j=1}^{n} v_j \left(g_{ij} \sin\theta_{ij} - b_{ij} \cos\theta_{ij} \right) \tag{2.1b}$$

$$P_{k:i\to j} = v_i v_j \left(g_{ij} \cos\theta_{ij} + b_{ij} \sin\theta_{ij} \right) - g_{ij} v_i^2 \tag{2.1c}$$

$$Q_{k:i\to j} = v_i v_j \left(g_{ij} \sin\theta_{ij} - b_{ij} \cos\theta_{ij} \right) + b_{ij} v_i^2 \tag{2.1d}$$

where P_i and Q_i are the active and reactive power injections at bus i; $P_{k:i\to j}$ and $Q_{k:i\to j}$ are the k-th active and reactive power flows from bus i to j; $\theta_{ij} = \theta_i - \theta_j$ is the phase angle difference between bus i and j.

Given the measurements, the AC-SE is solved by the following weighted least-square problem using an iterative algorithm, such as the Gauss-Newton method [38]:

$$\min_{\hat{\boldsymbol{\nu}}} J(\hat{\boldsymbol{\nu}}) = (\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{\nu}}))^T \cdot \boldsymbol{R}^{-1} \cdot (\boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{\nu}})) \tag{2.2}$$

where $\hat{\boldsymbol{\nu}}$ is the estimated state. Furthermore, the bad data detection (BDD) at the control center detects any measurement error that violates a Gaussian prior. Given $\hat{\boldsymbol{\nu}}$, the residual vector is calculated as $\boldsymbol{r} = \boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{\nu}})$ and the residual is represented as $\gamma(\boldsymbol{z}) = \|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{r}\|_2^2$, which approximately follows χ^2 distribution with degree of freedom (DoF) $p-(2n+1)$ [38]. The threshold $\tau_\chi(\alpha)$ of the χ^2 detector can be defined probabilistically based on the desired False Positive Rate (FPR) $\alpha \in (0,1)$ by the system operator [38]:

$$\int_{\tau_\chi(\alpha)}^{\infty} g(u)du = \alpha \tag{2.3}$$

where $g(u)$ is the p.d.f of the χ^2 distribution and α is usually set as 1%-5%. Consequently, the BDD detector can be designed as:

$$\mathcal{D}_{BDD}(\boldsymbol{z}) = \begin{cases} 1 & \gamma(\boldsymbol{z}) \geq \tau_\chi(\alpha) \\ 0 & \gamma(\boldsymbol{z}) < \tau_\chi(\alpha) \end{cases}$$

where 1 represents the “with attack” alarm and 0 represents the “without attack” alarm.

2.1.3 Attack Assumptions

With the emerging implementation of information and communication techniques, standard protocols, such as Modbus, can be vulnerable to FDI attacks. It has been shown that an FDI attack $\boldsymbol{z}_a = \boldsymbol{z} + \boldsymbol{a}$ can bypass the BDD if $\boldsymbol{a} = \boldsymbol{h}(\boldsymbol{\nu} + \boldsymbol{c}) - \boldsymbol{h}(\boldsymbol{\nu})$ where $\boldsymbol{c}$ is the attack vector on the state vector. In this case, the contaminated measurement becomes $\boldsymbol{z}_a = \boldsymbol{h}(\boldsymbol{\nu}+\boldsymbol{c})+\boldsymbol{e}$ whose residual follows the same χ^2 distribution as the legit measurement $\boldsymbol{z}$.

To successfully launch FDI attacks, we assume the attacker’s abilities as follows.

Assumption 1: The attackers can access all measurements and are aware of the admittance and topology of the grid to build $\boldsymbol{h}(\cdot)$. The exfiltration can be achieved by data-driven algorithms [61–63, 130]. However, the duration of data collection is much longer than a single state estimation time, implying that the attacker cannot immediately know the exact value of reactance changes [73]. Meanwhile, attackers are also aware of the exact state or estimation of the state from previous measurements [42, 43].

Assumption 2: The attackers can modify or replace all the eavesdropped measurements to achieve their purposes. However, since large instant measurement changes may violate the temporal trends of the grid measurements and be detected [56,131], the attack strength $\|\boldsymbol{a}\|_2$ is assumed to be small.

Assumptions 1-2 require the attacker's efforts to gain sufficient knowledge on the grid topology and operational conditions, which may not be easy in practice. However, we assume a strong attack ability and study the defence algorithm against general and unpredictable FDI attacks.

2.1.4 Moving Target Defence

By using the D-FACTS devices, the system operator can proactively change the reactances to keep invalidating the attacker's knowledge on $\boldsymbol{h}(\cdot)$:

$$\boldsymbol{h}_{\boldsymbol{x}}(\cdot) \xrightarrow{\text{D-FACTS}} \boldsymbol{h}_{\boldsymbol{x}'}(\cdot)$$

where $\boldsymbol{x}' = \boldsymbol{x} + \Delta\boldsymbol{x}$ is the reactance after activating the D-FACTS devices. As illustrated in Fig. 2.1, the channels of D-FACTS devices are encrypted and MTD is implemented with a period shorter than the reconnaissance time of the attacker (see Assumption 1). In addition, the reactances changed by the D-FACTS devices are physically limited:

$$-\tau\boldsymbol{x}_i \leq \Delta\boldsymbol{x}_i \leq \tau\boldsymbol{x}_i, \quad i \in \mathcal{E}_D \tag{2.4a}$$

$$\Delta\boldsymbol{x}_i = 0, \quad i \in \mathcal{E} \setminus \mathcal{E}_D \tag{2.4b}$$

where $\boldsymbol{x}_i$ is the reactance of the ith branch; τ represents the maximum perturbation ratio of D-FACTS devices. Typical values of τ are reported as $20\% - 50\%$ in the literature [68–70,73]; $\mathcal{E}_D$ represents the set of branches equipped with the D-FACTS devices. After implementing MTD, the residual vector becomes $\boldsymbol{r}_a' = \boldsymbol{h}'(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x} + \boldsymbol{c}) - \boldsymbol{h}(x) + \boldsymbol{e}$ which may no longer follow the χ^2 distribution of the legit measurement and hence trigger the BDD.

2.1.5 Model Simplification for MTD Design

To design the MTD against FDI attacks, most of the literature relies on DC or simplified AC power system models [68–71, 73, 80, 132] and then verifies the performance on the full AC model. Here, we adopt the simplified AC model based on the linearized measurement equation. Compared with the DC model, the simplified AC model can reflect different state values with branch resistance also considered.

In detail, the first-order Taylor expansion can be established around a stationary state $\boldsymbol{\nu}_0$ [72]:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{\nu}_0) + \boldsymbol{J}_{\boldsymbol{\nu}_0}(\boldsymbol{\nu} - \boldsymbol{\nu}_0) + \boldsymbol{e} \tag{2.5}$$

where the Jacobian matrix of $\boldsymbol{h}(\cdot)$ is found with respect to $\boldsymbol{\nu}_0$ as $\boldsymbol{J}_{\boldsymbol{\nu}_0} = \left[\frac{\partial \boldsymbol{h}_k}{\partial \boldsymbol{\nu}_i} \Big|_{\boldsymbol{\nu}=\boldsymbol{\nu}_0} \right]_{i,k}$. The state $\boldsymbol{\nu}_0$ can be simulated from security constrained AC-OPF [129] around the estimated active and reactive loads before the real-time operation. Alternatively, the states estimated from the previous measurements or a flat state [80, 81] can also be used. Following the recent literature on MTD [68, 70, 80], we consider the FDI attacks on the voltage phase angle and derive the defence strategies according to the power flow measurements at each branch. Therefore, the complete Jacobian matrix of the power flow measurements with respect to the phase angles can be rewritten as follows.

$$\boldsymbol{J}_{\boldsymbol{\theta_0}} = \left[\frac{\partial \boldsymbol{P}_{k:i\to j}}{\partial \boldsymbol{\theta}_i} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \right]_{k=1,\cdots,m} = -\boldsymbol{V} \cdot \boldsymbol{G} \cdot \boldsymbol{A}_r^{\sin} + \boldsymbol{V} \cdot \boldsymbol{B} \cdot \boldsymbol{A}_r^{\cos} \tag{2.6}$$

where $\boldsymbol{V} = \operatorname{diag}\left((\boldsymbol{C}_f \boldsymbol{v}) \circ (\boldsymbol{C}_t \boldsymbol{v})\right)$; $\boldsymbol{G} = \operatorname{diag}(\boldsymbol{g})$; $\boldsymbol{B} = \operatorname{diag}(\boldsymbol{b})$; $\boldsymbol{A}_r^{\sin} = \operatorname{diag}(\sin \boldsymbol{A}\boldsymbol{\theta_0})\boldsymbol{A}_r$; and $\boldsymbol{A}_r^{\cos} = \operatorname{diag}(\cos \boldsymbol{A}\boldsymbol{\theta_0})\boldsymbol{A}_r$. Moreover, $\boldsymbol{C}_f$ and $\boldsymbol{C}_t$ are the ‘from’ and ‘to’ -side incidence matrices; $\boldsymbol{A}_r$ is the reduced incidence matrix by removing the column representing the reference bus from the incidence matrix $\boldsymbol{A}$. To simplify the notation, we omit the subscript $\boldsymbol{\theta}_0$ in $\boldsymbol{J}_{\boldsymbol{\theta_0}}$ in the following discussion.

According to Assumption 2, as the attack strength is limited, the attack vector can also be linearized around $\boldsymbol{\theta}_0$ as [80]:

$$\boldsymbol{a} = \boldsymbol{h}(\boldsymbol{\theta}_0 + \boldsymbol{c}) - \boldsymbol{h}(\boldsymbol{\theta}_0) = \boldsymbol{J}\boldsymbol{c} \tag{2.7}$$

We design the MTD algorithm based on the simplified AC model (2.5)-(2.7) using active power flow measurements. The proposed MTD will be applied to the original AC model (2.1)-(2.2) in the simulation.

2.2 Analysis on MTD Effectiveness

In this section, we first extend the concept of complete MTD in the literature from DC model to simplified AC model. We then define the MTD effectiveness in a probabilistic way and illustrate the need for a new metric on effective MTD design in a noisy environment.

2.2.1 Complete MTD

Let $\boldsymbol{H}$ and $\boldsymbol{H}'$ be the DC measurement matrices. Under the noiseless condition, the *complete MTD* can be designed to detect any FDI attack by keeping the composite matrix $[\boldsymbol{H}, \boldsymbol{H}']$ full column rank [68–71]. If the full rank condition cannot be achieved due to the sparse grid topology (e.g. $m < 2n$) or limited number of D-FACTs devices, a *max-rank incomplete MTD* can be designed to minimize the attack space. As the rank of the composite matrix is maximized under both complete and incomplete conditions, we refer to the MTD strategies in [68–71] as *max-rank MTD*.

To better define the problem, we extend the concept of complete and incomplete MTDs from the DC model to the simplified AC models in the following proposition:

Proposition 1. *The power system modeled by* (2.5) *is with complete configuration against the FDI attack modeled by* (2.7) *only if* $m \geq 2n$ *where* m *and* n *are the number of branches and the number of non-reference buses, respectively.*

The proof can be found in Appendix B.1.

As stated by Proposition 1, to have a complete configuration $\text{rank}([\boldsymbol{J}_N, \boldsymbol{J}_N']) = 2n$, the number of branches should be at least one time larger than the number of non-reference buses. In addition, the max-rank incomplete MTD with $\text{rank}([\boldsymbol{J}_N, \boldsymbol{J}_N']) = m$ can be designed for the grid with incomplete configuration. In the following discussions, we refer to the grid that can achieve complete MTD under certain topology and D-FACTS device deployment as *complete configuration*, otherwise as *incomplete configuration*.

2.2.2 β-Effective MTD

Following (2.5), denote $\boldsymbol{z} \triangleq \boldsymbol{z} - \boldsymbol{h}(\boldsymbol{\theta}_0)$ and $\boldsymbol{\theta} \triangleq \boldsymbol{\theta} - \boldsymbol{\theta}_0$. For the new system equation $\boldsymbol{z} = \boldsymbol{J}\boldsymbol{\theta} + \boldsymbol{e}$, the residual vector of the χ^2 detector can be written as $\boldsymbol{r} = \boldsymbol{S}(\boldsymbol{J}\boldsymbol{\theta} + \boldsymbol{e}) = \boldsymbol{S}\boldsymbol{e}$ where $\boldsymbol{S} = \boldsymbol{I} - \boldsymbol{J}(\boldsymbol{J}^T\boldsymbol{R}^{-1}\boldsymbol{J})^{-1}\boldsymbol{J}^T\boldsymbol{R}^{-1}$ is the weighted orthogonal projector on $\text{Ker}(\boldsymbol{J}^T)$. The residual $\gamma = \|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}\boldsymbol{e}\|_2^2$ follows the χ^2 distribution with DoF $m-n$. Referring to the simplified attack model (2.7), the residual vector after MTD under attack can be written as $\boldsymbol{r}_a' = \boldsymbol{S}'(\boldsymbol{J}\boldsymbol{c} + \boldsymbol{e})$ where $\boldsymbol{S}' = \boldsymbol{I} - \boldsymbol{J}'(\boldsymbol{J}'^T\boldsymbol{R}^{-1}\boldsymbol{J}')^{-1}\boldsymbol{J}'^T\boldsymbol{R}^{-1}$. As $\boldsymbol{a}$ is usually not in $\mathscr{J}'$ and $\boldsymbol{r}_a'$ is biased from zero, the residual $\gamma_a' = \|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}'(\boldsymbol{J}\boldsymbol{c} + \boldsymbol{e})\|_2^2$ follows the non-central χ^2 distribution, i.e. $\gamma_a' \sim \chi^2_{m-n}(\lambda)$ with non-centrality parameter $\lambda = \|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}'\boldsymbol{J}\boldsymbol{c}\|_2^2$ [133]. Meanwhile, the mean and variance of the distribution are given as $\mathbf{E}(\gamma_a') = m - n + \lambda$ and $\mathbf{Var}(\gamma_a') = 2(m - n + 2\lambda)$, respectively. For a clear presentation, the matrices are normalized with respect to the measurement noises, e.g., $\boldsymbol{J}_N = \boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{J}$ and $\boldsymbol{a}_N = \boldsymbol{J}_N\boldsymbol{c}$. More details on the transformation can be found in Appendix C.1.

It is clear that when a noisy environment is considered, deterministic criteria can no longer be used to describe the effectiveness of MTD. A probabilistic criterion is hence defined. Following (2.3), for any given attack vector $\boldsymbol{a}$, we define an MTD as *β-effective* (β-MTD in short) if the following inequality is satisfied:

$$f(\lambda) = \int_{\tau_\chi(\alpha)}^{\infty} g_\lambda(u)du \geq \beta \tag{2.8}$$

where $g_\lambda(u)$ is the p.d.f. of non-central χ^2 distribution and $\beta \in (0,1)$ is a desired detection rate. When λ increases from 0, the detection probability on $\boldsymbol{a}$ also increases as the mean and variance increase [134]. Therefore, for a given β, there exists a minimum λ such that (2.8) is satisfied. This minimum λ is defined as critical and denoted as $\lambda_c(\beta)$.

Consequently, the rank conditions in [68–71, 80] cannot guarantee the detection performance, as they are not directly linked with the increase of λ to have β-MTD. Fig. 2.2 illustrates the c.d.f. of γ' on a random FDI attack using max-rank MTDs in a case-14 system. Without using MTD, the detection rate is 5% which is consistent with the FPR. To have a high detection rate, e.g., $\beta = 95\%$, it is desirable to sufficiently shift the distribution as shown by the blue curve. The max-rank MTDs can shift the c.d.f. positively, but there is no guarantee on how much of such shift can be achieved and whether it leads to the desired detection rates. This finding clearly calls for a new design of MTD algorithm in a noisy environment.

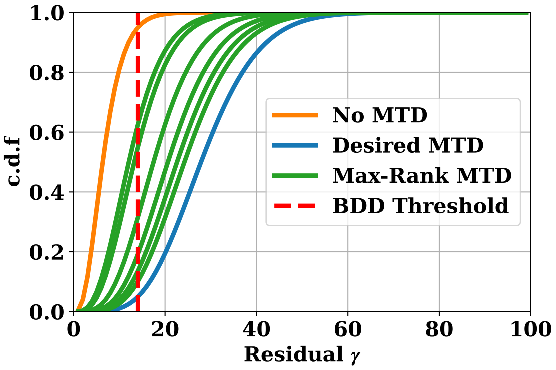


Figure 2.2: Illustration of attack detection probability on IEEE case-14 system based on simplified AC model (2.5)-(2.7). The more positively the c.d.f. is shifted, the higher averaged detection rate can be achieved.

Moreover, as numerically shown by [132], not all attacks can be detected by the MTD with high detection rate. Therefore, we theoretically introduce the following necessary condition to have β-MTD which can be seen as the limitation of MTD against FDI attacks with small attack strength.

Proposition 2. *An MTD is β-effective only if $\|\boldsymbol{a}_N\|_2 \geq \sqrt{\lambda_c(\beta)}$.*

The proof can be found in Appendix B.2.

Proposition 2 can be further analyzed on $\boldsymbol{a}$ to have $\|\boldsymbol{a}\|_2 \geq \sigma_{min}\sqrt{\lambda_c(\beta)}$ with $\sigma_{min} =$

$\min_i\{\sigma_1, \sigma_2, \ldots, \sigma_m\}$. This implies that β-MTD can be achieved only if the ratio between attack strength and measurement noise is higher than a certain value, which verifies the numerical results in [132].

2.2.3 Max MTD

While Proposition 2 establishes the theoretical limit on the detection probability for any given attack strength, in practice, the constraints on D-FACTS devices (2.4a)-(2.4b) further restricts such limit. In this context, the maximum detection rate on a known attack vector $\boldsymbol{a}_N$, with the limits of the D-FACTS devices considered, can be found by the *max-MTD* algorithm:

$$\begin{aligned} \max_{\Delta \boldsymbol{x}} \quad & \|\boldsymbol{S}'_N \boldsymbol{a}_N\|_2^2 \\ \text{subject to} \quad & (2.4\text{a}) - (2.4\text{b}) \end{aligned} \tag{2.9}$$

In practice, it is impossible to design $\Delta\boldsymbol{x}$ to achieve a certain $\lambda_c(\beta)$ in advance as $\boldsymbol{a}_N$ cannot be known. Nonetheless, max-MTD can be regarded as the performance upper-bound for any MTD strategy with the same placement and perturbation limit.

2.3 Robust MTD Algorithms

In this section, we start by establishing the concept of robust MTD and its mathematical formulation. Then the robust MTD algorithms are formulated for the grid with complete and incomplete configurations, respectively.

2.3.1 Definition and Problem Formulation

Instead of considering the average detection rate, this chapter defines the robust MTD that can maximize the worst-case detection rate against all possible attacks. First, we define the weakest point for a given MTD design as follows.

Definition 1. *Given $\Delta\boldsymbol{x}$ and the corresponding pair of subspaces $(\mathcal{J}_N, \mathcal{J}'_N)$, the weakest point of $(\mathcal{J}_N, \mathcal{J}'_N)$ is defined as a unitary element $\boldsymbol{j}^*_N \in \mathcal{J}_N$ such that $\lambda(\Delta\boldsymbol{x}, \boldsymbol{j}^*_N) \leq$*

$\lambda(\Delta\boldsymbol{x}, \boldsymbol{j}_N)$ *for* $\forall \boldsymbol{j}_N \in \mathcal{J}_N$, $\|\boldsymbol{j}_N\|_2 = 1$. *The worst-case detection rate for attack strength* $\|\boldsymbol{a}_N\|_2 = |a| \neq 0$ *is defined as* $f(\lambda_{min})$ *with* $\lambda_{min} = \lambda(\Delta\boldsymbol{x}, a\boldsymbol{j}_N^*)$.

According to the Definition 1, the weakest point in $(\mathcal{J}_N, \mathcal{J}'_N)$ satisfies $|a|\|\boldsymbol{S}'_N\boldsymbol{j}_N^*\|_2 \leq |a|\|\boldsymbol{S}'_N\boldsymbol{j}_N\|_2$, $\forall \boldsymbol{j}_N \in \mathcal{J}_N, \|\boldsymbol{j}_N\|_2 = 1, a \neq 0$. Let $\boldsymbol{a}_N^* = a\boldsymbol{j}_N^*$ and $\boldsymbol{a}_N = a\boldsymbol{j}_N$, the detection rate on $\boldsymbol{a}_N^*$ is the lowest among all attacks with the same strength as $\|\boldsymbol{S}'_N\boldsymbol{a}_N^*\|_2 \leq \|\boldsymbol{S}'_N\boldsymbol{a}_N\|_2, \forall \boldsymbol{a}_N \in \mathcal{J}_N, \|\boldsymbol{a}_N\|_2 = |a| \neq 0$. Note that the weakest point may not be unique, but all of them have the same worst-case detection rate.

Based on the definition of MTD weakest point, the following robust max-min optimization problem can be formulated:

$$\begin{aligned} \max\nolimits_{\Delta\boldsymbol{x}} \min\nolimits_{\|\boldsymbol{a}_N\|_2=1, \boldsymbol{a}_N \in \mathcal{J}_N} \quad & \|\boldsymbol{S}'_N\boldsymbol{a}_N\|_2^2 \\ \text{subject to} \quad & (2.4\text{a}) - (2.4\text{b}) \end{aligned} \tag{2.10}$$

The inner problem $\min_{\|\boldsymbol{a}_N\|_2=1, \boldsymbol{a}_N \in \mathcal{J}_N} \|\boldsymbol{S}'_N\boldsymbol{a}_N\|_2^2$ is the mathematical formulation of the weakest point in Definition 1 which is maximized over the outer programming. From a game-theoretic point of view, we can present this setting as an intelligent attacker aims to develop an FDI attack with the highest probability to bypass BDD and the system operator tries to improve his/her defence strategy against this intelligent attacker.

In the following sections, we will show that the two-stage problem (2.10) can be reduced into a single-stage minimization problem by analytically representing the weakest point using the principal angles between $\mathcal{J}_N$ and $\mathcal{J}'_N$.

2.3.2 Robust MTD for the Grid with Complete Configuration

Similar to the one-dimensional case where the angle between two unitary vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is defined as $\cos\theta = \boldsymbol{v}^T\boldsymbol{u}$, the minimal angle between subspaces $\mathcal{J}_N, \mathcal{J}'_N \subseteq \mathbb{R}^p$ is defined as $0 \leq \theta_1 \leq \pi/2$ [135]:

$$\cos\theta_1 = \max_{\substack{\boldsymbol{u} \in \mathcal{J}_N, \boldsymbol{v} \in \mathcal{J}'_N \\ \|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2 = 1}} \boldsymbol{u}^T\boldsymbol{v} = \boldsymbol{u}_1^T\boldsymbol{v}_1 \tag{2.11}$$

where θ_1 is the minimal principal angle; $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ are the first principal vectors. Referring to (2.11), the following proposition specifies that the weakest point with the lowest detection rate of $(\mathcal{J}_N, \mathcal{J}'_N)$ is the first principal vector $\boldsymbol{u}_1$ associated with the minimal principal angle θ_1.

Proposition 3. *Given a pair of $(\mathcal{J}_N, \mathcal{J}'_N)$, the minimum non-centrality parameter under attack strength $\|\boldsymbol{a}_N\|_2 = |a| \neq 0$ is $\lambda_{min} = a^2 \sin^2 \theta_1$. Meanwhile, λ_{min} is achieved by attacking the first principal vector $\boldsymbol{u}_1$ of $\boldsymbol{J}_N$.*

The proof can be found in Appendix B.3.

When $\theta_1 = \pi/2$, Proposition 3 implies that the minimum non-centrality parameter is equal to a^2. As two subspaces are orthogonal if $\theta_1 = \pi/2$, Proposition 3 is consistent with the maximum detection probability stated in Theorem 1 of [73].

In addition, as $\sin \cdot$ is monotonically increasing in $[0, \pi/2]$, Proposition 3 demonstrates that the two-stage problem (2.10) can be equivalently solved by one-stage maximization:

$$\begin{aligned} \max\nolimits_{\Delta \boldsymbol{x}} \quad & \theta_1 \\ \text{subject to} \quad & (2.4\text{a}) - (2.4\text{b}) \end{aligned} \tag{2.12}$$

To analytically represent θ_1, a sequence of principal angles $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ can be defined iteratively by finding the orthonormal basis of $\mathcal{J}_N$ and $\mathcal{J}'_N$ such that for $i = 2, \ldots, n$ [135]:

$$\cos \theta_i = \max_{\substack{\boldsymbol{u} \in \mathcal{J}_{N,i}, \boldsymbol{v} \in \mathcal{J}'_{N,i} \\ \|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2 = 1}} \boldsymbol{u}^T \boldsymbol{v} = \boldsymbol{u}_i^T \boldsymbol{v}_i \tag{2.13}$$

where $\mathcal{J}_{N,i} = \boldsymbol{u}_{i-1}^{\perp} \cap \mathcal{J}_{N,i-1}$ and $\mathcal{J}'_{N,i} = \boldsymbol{v}_{i-1}^{\perp} \cap \mathcal{J}'_{N,i-1}$.

Θ can be separated into three parts. Let $\Theta_1 = \{\theta_i | \theta_i = 0\}$, $\Theta_2 = \{\theta_i | 0 < \theta_i < \pi/2\}$, and $\Theta_3 = \{\theta_i | \theta_i = \pi/2\}$ with cardinality equal to k, r, and l, respectively, and $n = k+r+l$. The corresponding vectors $\boldsymbol{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\}$ and $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ are called principal vectors, which are the orthonormal basis of $\mathcal{J}_N$ and $\mathcal{J}'_N$, respectively. Similarly, $\boldsymbol{U}$ and $\boldsymbol{V}$ can also be separated into $\boldsymbol{U}_1, \boldsymbol{V}_1, \cdots$. Specifically, $\boldsymbol{U}_1 = \boldsymbol{V}_1 = \mathcal{J}'_N \cap \mathcal{J}_N$ represents the intersection subspace of dimension k and l is the dimension of orthogonality. Furthermore, it is proved that there always exist semi-orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ for any $\mathcal{J}_N$ and $\mathcal{J}'_N$ such that the bi-orthogonality is satisfied [136]:

$$\boldsymbol{U}^T\boldsymbol{V} = \mathrm{diag}([\cos\theta_1, \cos\theta_2, \ldots, \cos\theta_n]) = \Gamma \tag{2.14}$$

Since the orthogonal projector is uniquely defined [135] and also by (2.14), rewriting $\boldsymbol{P}_N = \boldsymbol{U}\boldsymbol{U}^T$ and $\boldsymbol{P}'_N = \boldsymbol{V}\boldsymbol{V}^T$ gives

$$\boldsymbol{P}_N\boldsymbol{P}'_N = \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{U}\Gamma\boldsymbol{V}^T \tag{2.15}$$

Eq. (2.15) is the truncated singular value decomposition (t-SVD) on $\boldsymbol{P}_N\boldsymbol{P}'_N$ where the diagonal matrix Γ contains the first n largest singular values of $\boldsymbol{P}_N\boldsymbol{P}'_N$, and $\boldsymbol{U}$ and $\boldsymbol{V}$ are the first (left- and right-hand) n singular vectors of $\boldsymbol{P}_N\boldsymbol{P}'_N$. As $\sigma(\boldsymbol{P}_N\boldsymbol{P}'_N) = \{\mathbf{1}_k, \cos\theta_{k+i}(i = 1, \ldots, r), \mathbf{0}_{k+r+i}(i = 1, \ldots, l), \mathbf{0}_{n+i}(i = 1, \ldots, m-n)\}$, this t-SVD is an exact decomposition of $\boldsymbol{P}_N\boldsymbol{P}'_N$.

Based on the t-SVD, Algorithm 1 is proposed to find the weakest point and the worst-case detection rate. For the grid with complete configuration, the composite matrix can be full column rank so that $k = 0$. Line 6 outputs the weakest point $\boldsymbol{u}_1$ while line 9 outputs the empty intersection subspace. The worst-case detection rate is calculated according to Proposition 3 in line 7. Practically, once the MTD strategy is determined, the weakest point $\boldsymbol{u}_1$ of this strategy can be directly spotted. Therefore, the system operator can evaluate the worst-case detection rate with respect to a maximum tolerable attack strength $|a|$.

The t-SVD (2.15) also results in a solvable reformulation of (2.12). The worst-case detection rate can be maximised by the *robust MTD* algorithm for the grid with complete configuration as follows:

$$\begin{aligned} \min\nolimits_{\Delta\boldsymbol{x}} \quad & \|\boldsymbol{P}_N\boldsymbol{P}'_N\|_2 \\ \text{subject to} \quad & (2.4\text{a}) - (2.4\text{b}) \end{aligned} \tag{2.16}$$

where the property $\|\boldsymbol{P}_N\boldsymbol{P}'_N\|_2 = \sigma_{\max}(\boldsymbol{P}_N\boldsymbol{P}'_N) = \cos(\theta_1)$ is used and $\|\boldsymbol{P}_N\boldsymbol{P}'_N\|_2 \in [0, 1]$.

Algorithm 1: Find the Weakest Point(s) and the Worst-Case Detection Rate

```
Input  : grid topology $\mathcal{G}(\mathcal{N}, \mathcal{E})$, reactance perturbation $\Delta\boldsymbol{x}$, and attack strength
         $|a|$
Output: weakest point $\boldsymbol{u}_{k+1}$, intersection subspace $\boldsymbol{U}_1$, and worst-case detection
         rate $f_{min}$
Construct the pre- and post- MTD measurement matrices $\boldsymbol{J}_N$ and $\boldsymbol{J}'_N$,
  respectively;
Find the orthogonal projectors $\boldsymbol{P}_N$ and $\boldsymbol{P}'_N$ on $\boldsymbol{J}_N$ and $\boldsymbol{J}'_N$. Then do t-SVD (2.15);
$rank = \text{rank}([\boldsymbol{J}_N, \boldsymbol{J}'_N])$; /* Rank of the composite matrix.            */
$k = 2n - rank$; /* The dimension of $\mathcal{J}'_N \cap \mathcal{J}_N$.                        */
$\cos(\theta_{k+1}) = \Gamma(k+1, k+1)$;
$\boldsymbol{u}_{k+1} = \boldsymbol{U}(k+1, k+1)$; /* The weakest point in $\mathcal{J}_N \setminus (\mathcal{J}'_N \cap \mathcal{J}_N)$.   */
$f_{\min} = f(a^2 \sin^2(\theta_{k+1}))$; /* The worst-case detection rate in
    $\mathcal{J}_N \setminus (\mathcal{J}'_N \cap \mathcal{J}_N)$.                                            */
if $rank = 2n$ then
    $\boldsymbol{U}_1 = \varnothing$; /* Complete MTD configuration.                         */
else
    $\boldsymbol{U}_1 = \boldsymbol{U}(:, 1 : k)$; /* Incomplete MTD configuration.                */
end
```

Remark 1. *The robust MTD algorithm* (2.16) *requires sufficient placement of D-FACTS devices (as a planning stage problem) to guarantee $k = 0$, e.g., using the 'D-FACTS placement for the complete MTD' algorithm in [70].*

2.3.3 Robust MTD for the Grid with Incomplete Configuration

The robust MTD in (2.16) is not tractable for power system with incomplete MTD configuration. As $k \neq 0$, $\theta_1 \equiv 0$ and $\|\boldsymbol{P}_N \boldsymbol{P}'_N\|_2 \equiv 1$ no matter how $\Delta\boldsymbol{x}$ is designed. Fig. 2.3 shows a three-dimensional incomplete-MTD case. The attack $\boldsymbol{a}_N$ in green shows a random attack attempt with non-zero λ. However, the weakest point $\text{Col}(\boldsymbol{u_1})$ is not trivial. As the attacker can possibly target $\text{Col}(\boldsymbol{u}_1)$, the worst-case detection rate is constantly equal to FPR. In addition to θ_1, every attack in $\boldsymbol{U}_1$ is undetectable. The intersection can be regarded as the space of the weakest points, whose dimension is calculated as $k = 2n - \text{rank}([\boldsymbol{J}_N, \boldsymbol{J}'_N]) \neq 0$. Therefore, the smallest non-zero principal angle (which also corresponds to the weakest point in $\mathcal{J}_N \setminus (\mathcal{J}'_N \cap \mathcal{J}_N)$) can be found as θ_{k+1} in line 5 of Algorithm 1 with the minimum detection rate calculated in line 7. Meanwhile, $\boldsymbol{U}_1$,

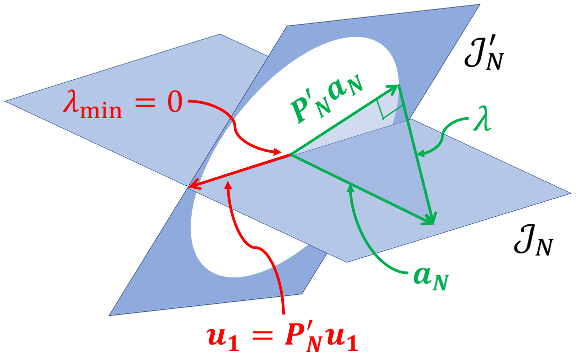


Figure 2.3: An illustration on the grid with incomplete configuration, $\mathcal{J}_N, \mathcal{J}'_N \subset \mathbb{R}^3$. In this three-dimension example, $\mathcal{J}_N$ and $\mathcal{J}'_N$ are two-dimensional subspaces of $\mathbb{R}^3$. By visualization, there always exists an intersection as long as the columns of $\mathcal{J}_N$ and $\mathcal{J}'_N$ span the corresponding two-dimensional subspaces. By definition, their minimum principal angle is always 0.

corresponding to the subspace that cannot be detected, is calculated in line 11.

To solve the intractable problem, the following design principles are considered which can improve the robust performance of MTD with incomplete configuration:

Principle 1: Minimize k, the dimension of the intersection.

Principle 2: The attacker shall not easily attack on the intersection subspace $\boldsymbol{U}_1$ by chance.

Principle 3: Maximize θ_{k+1}, the minimum non-zero principal angle in $(\mathcal{J}_N, \mathcal{J}'_N)$.

Each of the principles is discussed as follows.

Principle 1: The idea of Principle 1 is to minimize the attack space that can never be detected by MTD so that the probability of detectable FDI attacks increases. Minimizing k is a planning stage problem as the rank of the composite matrix is almost not related to the perturbation amount of the D-FACTS devices once they have been deployed [69]. In this chapter, we propose a new D-FACTS device placement algorithm to achieve the minimum k. Compared with the existing work [68–70], our algorithm uses the BLOSSOM algorithm [137] to find the maximum cardinality matching [138] of $\mathcal{G}(\mathcal{N}, \mathcal{E})$, which can reach all necessary buses with the smallest number of D-FACTS devices. More details are presented in Appendix C.2.

Principle 2: From the robust consideration, the following lemma is derived for the attacks targeting on the weakest point(s) for the grid with incomplete MTD configuration.

Lemma 1. *Let* $\boldsymbol{U} = (\boldsymbol{U}_1, \boldsymbol{U}_{2,3})$ *where* $\boldsymbol{U}_{2,3}$ *is the collection of columns in* $\boldsymbol{U}_2$ *and* $\boldsymbol{U}_3$*. Let* $\boldsymbol{a}_N = \boldsymbol{U}_1\boldsymbol{c}_1 + \boldsymbol{U}_{2,3}\boldsymbol{c}_{2,3}$ *with* $\boldsymbol{c}_1 \in \mathbb{R}^k$ *and* $\boldsymbol{c}_{2,3} \in \mathbb{R}^{r+l}$*. The detection rate on* $\boldsymbol{a}_N$ *does not depend on the value of* $\boldsymbol{c}_1$*.*

The proof can be found in Appendix B.4.

Although the attackers cannot immediately know the exact $\boldsymbol{x}'$ (Assumption 1), Lemma 1 suggests that the MTD algorithm should be designed such that the attackers cannot easily attack on $\boldsymbol{U}_1$ by chance. Specifically, considering the attack targeting a single state i, if $\text{Col}(\boldsymbol{J}_N(:,i)) \subseteq \boldsymbol{U}_1$, the single-state attack on the bus i can bypass the MTD while any attack involving bus i can be detected ineffectively. To avoid ineffective MTD on this attack, the following constraint is considered.

$$\|\boldsymbol{P}_N^i \boldsymbol{P}_N'\|_2 \geq \gamma_i, \quad \forall i \in \mathcal{N}^c \tag{2.17}$$

where $\boldsymbol{P}_N^i = \left(\boldsymbol{J}_N(:,i)^T \boldsymbol{J}_N(:,i)\right)^{-1} \boldsymbol{J}_N(:,i)\boldsymbol{J}_N(:,i)^T$ is the orthogonal projector on $\text{Col}(\boldsymbol{J}_N(:,i))$. $\mathcal{N}^c$ represents the index set of buses that are included in at least a loop[1] of $\mathcal{G}$. Since $\|\boldsymbol{P}_N^i \boldsymbol{P}_N'\| \in [0,1]$ and 1 is achieved when $\text{Col}(\boldsymbol{J}_N(:,i)) \subseteq \boldsymbol{U}_1$, the threshold γ_i can be set close but not equal to 1.

Notice that the constraint in (2.17) cannot eliminate the weakest point(s) nor improve the worst-case detection rate on $\boldsymbol{U}_1$, but it can restrict the attacker's knowledge on the weakest point(s). Rewriting λ as $\lambda = \|(\boldsymbol{I} - \boldsymbol{P}_N') \sum_{i=1}^n \boldsymbol{J}_N(:,i)\boldsymbol{c}(i)\|_2^2$, constraint (2.17) ensures that $(\boldsymbol{I} - \boldsymbol{P}_N')\boldsymbol{J}_N(:,i)\boldsymbol{c}(i) \neq 0$, $\forall i \in \mathcal{N}^c$. To have low MTD detection rate, the attacker has to coordinate the attack strength on at least two buses to have low λ which is only possible if $\boldsymbol{x}'$ is known. As long as the attacker cannot easily attack $\boldsymbol{U}_1$, the probability of having the worst case is low and the MTD strategy is still effective from a robust point of view.

Remark 2. *To fulfill constraint* (2.17)*, all buses in* $\mathcal{N}^c$ *should be incident to at least a*

[1] As proved by [76], if a bus is not included in any loop, attacks on this bus cannot be detected regardless of the MTD strategies.

branch equipped with D-FACTS devices, which can be achieved by the proposed D-FACTS devices placement algorithm in Appendix C.2.

Principle 3: Although the chance of the worst-case attack is minimized by Principle 1-2, it does not necessarily imply a high detection rate when $\boldsymbol{a}_N \notin \boldsymbol{U}_1$. Similarly to (2.12), the minimum non-zero principal angle θ_{k+1}, which represents the weakest point in subspace $\mathcal{J}_N \setminus (\mathcal{J}'_N \cap \mathcal{J}_N)$ should be maximised by

$$\begin{aligned} \min_{\Delta \boldsymbol{x}} \quad & \cos\theta_{k+1} \\ \text{subject to} \quad & (2.4\text{a}) - (2.4\text{b}), (2.17) \end{aligned} \tag{2.18}$$

where $\cos\theta_{k+1}$ is the $(k+1)$th largest singular value.

To our knowledge, there is no direct method to solve (2.18) as finding the singular value at a certain position requires solving the SVD of $\boldsymbol{P}_N \boldsymbol{P}'_N$ and locating the 1th to kth singular vectors. Therefore, we propose an iterative Algorithm 2 to solve (2.18). In line 1 of Algorithm 2, a warm start $\Delta \boldsymbol{x}^0$ is first found by minimizing the Frobenius norm $\|\cdot\|_F$, which is shown to be an upper bound to $\cos\theta_{k+1}$.

$$\begin{aligned} \min_{\Delta \boldsymbol{x}} \quad & \|\boldsymbol{P}_N \boldsymbol{P}'_N\|_F \\ \text{subject to} \quad & (2.4\text{a}) - (2.4\text{b}), (2.17) \end{aligned} \tag{2.19}$$

For a given warm-start perturbation $\Delta \boldsymbol{x}^0$, the intersection subspace $\boldsymbol{U}_1$ can be located by Algorithm 1. Denoting $\boldsymbol{U}_1(\Delta \boldsymbol{x}^0)$ as $\boldsymbol{U}_1^0$, the t-SVD (2.15) can be rewritten as

$$\begin{aligned} \boldsymbol{P}_N \boldsymbol{P}'_N = \quad & \left(\boldsymbol{U}_1^0, \boldsymbol{U}_{2,3}\right) \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \Gamma_{2,3} \end{pmatrix} \begin{pmatrix} \boldsymbol{V}_1^{0T} \\ \boldsymbol{V}_{2,3}^T \end{pmatrix} \\ = \quad & \boldsymbol{U}_1^0 \boldsymbol{U}_1^{0T} + \boldsymbol{U}_{2,3} \Gamma_{2,3} \boldsymbol{V}_{2,3}^T \end{aligned}$$

where $\boldsymbol{I}$ is the identity matrix of dimension k; $\Gamma_{2,3} = \text{diag}([\cos(\theta_{k+1}), \cdots, \cos(\theta_n)])$ with $\theta_{k+1} \neq 0$. Note that $\boldsymbol{U}_1^0 = \boldsymbol{V}_1^0 = \mathcal{J}'_N \cap \mathcal{J}_N$.

Algorithm 2: Robust MTD for the Grid with Incomplete Configuration

```
Input  : grid topology $\mathcal{G}(\mathcal{N},\mathcal{E})$, terminating tolerance $tol$, maximum iteration
         number $max\_ite$
Output: reactance perturbation $\Delta\boldsymbol{x}^1$
Find the warm start point $\Delta\boldsymbol{x}^0$ by solving (2.19);
Find the intersection subspace $\boldsymbol{U}_1^0$ by Algorithm 1;
/* iteration until convergence.                                    */
while $step < max\_ite$ do
    Find $\Delta\boldsymbol{x}^1$ by solving (2.20);
    Find the intersection subspace $\boldsymbol{U}_1^1$ by Algorithm 1;
    if $\|\boldsymbol{U}_1^1 - \boldsymbol{U}_1^0\|_2 \leq tol$ then
        break; /* converged.                                       */
    else
        $\boldsymbol{U}_1^0 := \boldsymbol{U}_1^1$;
    end
end
```

Therefore, the following optimization problem can be formulated to minimize $\cos\theta_{k+1}$:

$$\begin{aligned} \min\nolimits_{\Delta\boldsymbol{x}} \quad & \|\boldsymbol{P}_N\boldsymbol{P}_N' - \boldsymbol{U}_1^0\boldsymbol{U}_1^{0T}\|_2 \\ \text{subject to} \quad & (2.4\text{a}) - (2.4\text{b}), (2.17) \end{aligned} \tag{2.20}$$

Denoting the optimal value of (2.20) as $\Delta\boldsymbol{x}^1$, a new intersection subspace $\boldsymbol{U}_1^1 = \boldsymbol{U}_1(\Delta\boldsymbol{x}^1)$ can be located. As $\Delta\boldsymbol{x}^1$ is solved with fixed $\boldsymbol{U}_1^0$, $\boldsymbol{U}_1^1$ may not be the same as $\boldsymbol{U}_1^0$. After finding the new intersection subspace from $\Delta\boldsymbol{x}^1$, (2.20) can be iteratively solved until convergence, as shown by line 3-11 in Algorithm 2.

To sum up, Algorithm 2 limits the chance of attacking on $\mathcal{J}_N' \cap \mathcal{J}_N$ (Principal 1-2) and guarantees the worst-case detection rate in $\mathcal{J}_N \setminus (\mathcal{J}_N' \cap \mathcal{J}_N)$ (Principal 3 and (2.19)-(2.20)) for the grid with incomplete configuration.

2.3.4 Discussions on Full AC Model Design

In previous sections, we theoretically established the robust MTD algorithm based on the simplified AC model (2.5)-(2.7). There exists similar concept on the weakest point in the original AC settings (2.1)-(2.2). Let $\boldsymbol{h}'^{-1}(\cdot)$ represent the result of state estimation in (2.2). The estimated state on attacked measurement is written as $\hat{\boldsymbol{\nu}}_a' = \boldsymbol{h}'^{-1}(\boldsymbol{z}_a')$

and the residual is $\gamma_a' = \|\boldsymbol{R}^{-\frac{1}{2}}(\boldsymbol{z}_a' - \boldsymbol{h}'(\hat{\boldsymbol{\nu}}_a'))\|_2^2$. The weakest point can be defined as a unitary attack vector such that γ_a' is minimized. However, there are several obstacles to analytically writing its expression. Firstly, recall that $\boldsymbol{a} = \boldsymbol{h}(\boldsymbol{\nu}' + \boldsymbol{c}) - \boldsymbol{h}(\boldsymbol{\nu}')$ which is non-linearly dependent on the post-MTD state $\boldsymbol{\nu}'$ and the state attack vector $\boldsymbol{c}$. Note that $\boldsymbol{\nu}'$ is dependent on $\boldsymbol{x}'$ which cannot be determined in advance. Second, $\boldsymbol{h}'^{-1}(\cdot)$ requires an iterative update, such as the Gauss-Newton or Quasi-Newton algorithm. Although it is possible to reformulate AC-SE as semi-definite programming [139], it lacks of analytical solution in general. Third, it is difficult to define the concept of angles between subspaces defined by two functions $\boldsymbol{h}(\cdot)$ and $\boldsymbol{h}'(\cdot)$. Consequently, we theoretically derived the robust algorithm based on the simplified AC model and numerically verify the performance on AC-FDI attacks in simulation. We found out that the MTD designed by the sufficient separation between the subspaces between the real-time Jacobian matrices can provide effective detection in the full AC model.

2.4 Simulation

2.4.1 Simulation Set-ups

We test the proposed algorithms on IEEE benchmarks case-6, case-14, and case-57 from MATPOWER [140]. AC-OPF is solved using the Python package PYPOWER 5.1.15. and the nonlinear optimization problems are solved using the open source library SciPy. More simulation setups are given below.

Attack Pools and BDD Threshold

Firstly, we define the attack strength with respect to the noise level as:

$$\rho = \frac{\|\boldsymbol{a}\|_2}{\sqrt{\sum_i^m \sigma_i^2}} \tag{2.21}$$

We consider three types of attacks for the simplified AC model. 1). ***Worst-case attack*** where the attacker attacks on the weakest point $\boldsymbol{u}_{k+1}$ of a given MTD strategy according to Algorithm 1; 2). ***Single-state attack*** where the attacker only injects on single non-reference phase angle; and 3). ***Random attack*** where the attack vector $\boldsymbol{a}$ is randomly generated as follows. First, the number of attacked state $\|\boldsymbol{c}\|_0 = q$ is drawn uniformly from set $\{1, 2, \ldots, n\}$. $\boldsymbol{c}$ is then sampled from multivariate Gaussian distribution with q non-zero entries. Second, the attack vector is found as $\boldsymbol{a} = \boldsymbol{J}\boldsymbol{c}$ and rescaled by different $\rho = 5, 7, 10, 15, 20$ according to (2.21). To simplify the analysis, the measurement noise is set as $\sigma_i = 0.01 p.u., \forall i$ in all case studies. In this case, to have β-MTD, the necessary condition is $\rho \geq \sqrt{\lambda_c(\beta)/m}$ according to Proposition 2.

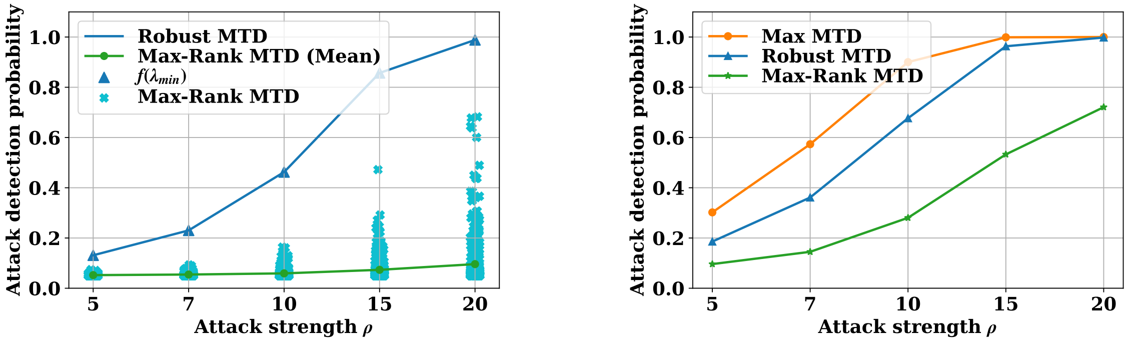
In the original AC model, the measurement consists of P_i, Q_i, $P_{k:i\to j}$, and $Q_{k:i\to j}$ (2.1), which are nonlinearly dependent on $\boldsymbol{\theta}$. Therefore, we randomly sample $\boldsymbol{c}$ from uniform distribution and classify $\boldsymbol{a} = \boldsymbol{h}'(\boldsymbol{\nu}' + \boldsymbol{c}) - \boldsymbol{h}'(\boldsymbol{\nu}')$ into one of the ranges $\{[5, 7)$, $[7, 10)$, $[10, 15)$, $[15, 20)$, $[20, 25)$, $[25, \infty)\}$ by (2.21).

We sample `no_load`=50 load conditions on a uniform distribution of the default load profile in MATPOWER [140] for each grid. We then set the D-FACTS devices using different MTD algorithms and simulate the real-time measurements. Under each load condition, we generate `no_attack`=200 attack attempts for each of the attack types. The BDD threshold $\tau_\chi(\alpha)$ is determined with $\alpha = 5\%$ FPR.

Metrics and Baselines

The key metric to evaluate the MTD detection performance is the true positive rate, also known as the attack detection probability (ADP), which is the ratio between the number of attacks that are detected by the MTD detector and the total number of attacks.

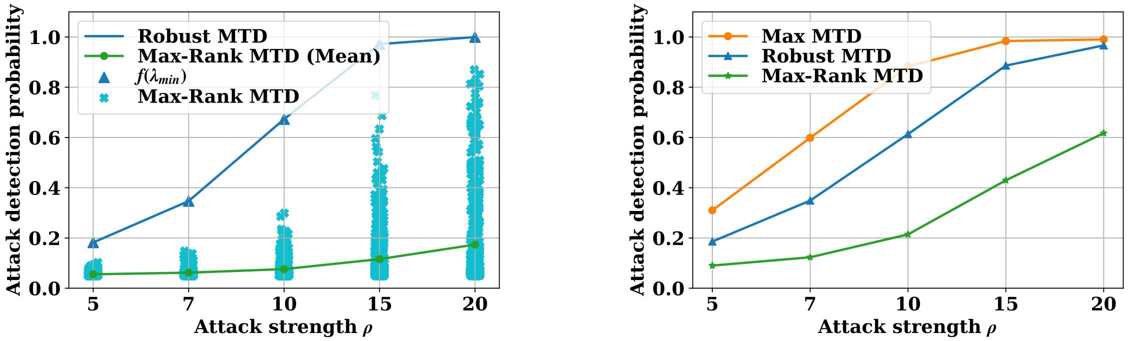
The max-rank MTD algorithm modified from [68–71] is compared as the baseline where reactances are randomly changed with $\mu_{min}\boldsymbol{x}_i \leq |\Delta \boldsymbol{x}_i| \leq \mu_{max}\boldsymbol{x}_i$. Note that each reactance is perturbed by $\mu_{min} > 0$ to fulfil the max-rank condition on the composite matrix. For each attempt of attack `no_attack`, we simulate `no_maxrank` $= 20$ MTDs of maximum rank to evaluate their average detection performance.



(a) Worst-Case Attack

(b) Random Attack

Figure 2.4: ADPs on simplified case-6 system.



(a) Worst-Case Attack

(b) Random Attack

Figure 2.5: ADPs on simplified case-14 system.

2.4.2 Verification of Theoretical Analysis on Simplified AC Model

In the first case study, we verify the theoretical analysis of robust MTD algorithms and demonstrate their effectiveness in the simplified AC model (2.5)-(2.7).

First, the ADPs of case-6 with complete configuration are illustrated in Fig. 2.4 for both worst-case attacks and random attacks. The reactances are changed with $\tau = 0.2$. Meanwhile, $\mu_{min} = 0.05$ and $\mu_{max} = 0.2$ in the max-rank MTD. In Fig. 2.4(a), the simulation result on the ADPs of robust MTD is the same as the theoretic detection rate $f(\lambda_{min})$ calculated by Proposition 3, which verifies the theoretic analysis and the design criteria. In addition, the robust MTD algorithm shows much higher ADPs than the max-rank MTD on the worst-case attack. Although the max-rank MTD's performance may approach the robust MTD in some cases, its average ADP is similar to the FPR as the

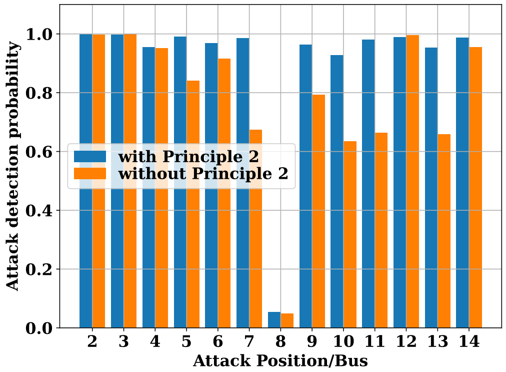


Figure 2.6: ADPs on single-state attacks of case-14 system.

worst-case performance cannot be explicitly considered under the noiseless setting.

In Fig. 2.4(b), the max MTD is added by solving (2.9) with the assumption that the attack vector $\boldsymbol{a}_N$ is known, which represents the performance upper-bound of any MTD design. As shown by Fig. 2.4(b), the robust MTD algorithm, not only guarantees the worst case condition, but also outperforms the max-rank MTD by 10%-45% on random attacks with different ρ. Moreover, the gap between robust MTD and max MTD algorithms is smaller than 25% and approaches to zero when $\rho \geq 15$. However, comparing Fig. 2.4(a) and Fig. 2.4(b), it is worth noting that the major improvement of robust MTD over max-rank MTD still lies in the worst-case attacks.

Fig. 2.5 investigates the performance on the case-14 system with incomplete configuration. By Algorithm 1, the minimum k is equal to 6 and the worst point in $\mathcal{J}_N \setminus (\mathcal{J}'_N \cap \mathcal{J}_N)$ is at $\boldsymbol{u}_7$. Assume that all branches are equipped with D-FACTS devices and the maximum perturbation ratio is set as $\tau = 0.2$. Although the detection rates on attacks in $\boldsymbol{U}_1$ are equal to α according to Lemma 1, the ADP on $\boldsymbol{u}_7$ is nonzero by implementing Algorithm 2 and increases as the strength of the attack increases. Similarly to Fig. 2.4(a), although the max-rank MTD algorithm can, by chance, give a high detection rate against the worst-case attack, its average detection rate is extremely low. In Fig. 2.5(b), the gap between the max MTD and the robust MTD is also small (5%-30%). The results demonstrate that robust design can also effectively improve the detection performance for the grid with incomplete configuration.

Table 2.1: Average ADPs on random AC-FDI attacks. Max-Rk represents the max-rank MTD, and Robust represents the robust MTD.

ρ	**case-6**		**case-14**		**case-57**	
	Max-Rk	Robust	Max-Rk	Robust	Max-Rk	Robust
$[5, 7)$	7.1%	13.7%	8.6%	18.1%	10.3%	30.3%
$[7, 10)$	12.6%	33.2%	14.4%	41.2%	15.2%	39.2%
$[10, 15)$	25.1%	67.3%	27.5%	63.1%	23.7%	55.9%
$[15, 20)$	44.5%	92.4%	43.4%	87.5%	36.0%	69.1%
$[20, 25)$	60.2%	98.2%	60.6%	94.5%	50.6%	81.6%

To further investigate on the weakest points in $\boldsymbol{U}_1$, we generate single-bus attack with $\rho = 10$ and record the ADPs in Fig. 2.6 with and without Principle 2 (2.17). First, attacks targeting bus-8 can only be detected by 5%. This is because bus-8 is a degree-one bus which is excluded by any loop. Second, with Principle 2 considered, the robust MTD can give more than 90% ADPs for all buses. In contrast, there are attacks against certain buses, e.g. bus-7, 10, 11, and 13 can be barely detected without Principle 2. Consequently, the simulation result verifies that Principle 2 can sufficiently reduce the chance of attacking the weakest points.

2.4.3 Simulation Results on Full AC Model

In this section, we verify the detection effectiveness of the proposed robust MTD algorithms on FDI attacks under the original AC settings (2.1)-(2.2).

Random Attack

Random attacks ADPs for the full-AC cases-6, case-14, and case-57 systems are summarized in Table 2.1. Similarly to studies on simplified AC models, the proposed robust algorithms can improve ADPs by 10%-40% compared to the max-rank algorithm. In particular, for cases with attack strength below 20, robust MTD can almost double the ADPs of max-rank MTD for all three systems. Therefore, the robust MTD designed by the principal angles between the subspaces of pre- and post- MTD Jacobian matrices are still effective on defending AC-FDI attacks. In addition, the attacks with larger attack strength are more likely to be detected while the detection probability for different systems under the same attack strength is slightly different due to their different load levels, parameters (e.g. the reactance to resistance ratios), and topologies. For instance, case-57 system is harder to detect as the ADPs in both max-rank and robust MTDs are lower than the case-6 and case-14 systems.

To confirm the detection performance, the residual distributions for the three systems are summarized in Fig. 2.7 where kernel density estimation is used to smooth the histograms. The result implies that the proposed algorithms can generalize well to AC-FDI attacks by sufficiently shifting the distribution positively, which is shown to be a key property on effective MTD with the measurement noise considered in Fig. 2.2. For each sub-figure, the max-rank MTD performs worse than the robust MTD on average as well.

Impact of Different Placements and Perturbation Ratios of D-FACTs Devices

Fig. 2.8 records the simulation results on AC random attacks under two different D-FACTS devices placements and four different perturbation ratio limits. In detail, 'all' represents perturbing all branches, whereas 'part' represents perturbing on branch- 2, 3, 4, 12, 15, 18, and 20, which is the outcome of the 'D-FACTS Devices Placement Algorithm' in Appendix C.2. The simulation result shows that $k = 6$ is achieved and all buses are covered except bus 8 in 'part' placement. As the maximum perturbation ratio is reported as 50% in literature [73], τ is set as 0.2, 0.3, 0.4, and 0.5. As a result, the grey curve in Fig. 2.8 is simulated in the same settings as the robust MTD in Table 2.1. When the number of D-FACTS devices is limited, although the minimum k is still met by Principle 1, the detection rate is significantly reduced. To attain a higher detection rate, the perturbation limit should be further increased. Notably, the dependence of ADP on different D-FACTS device placements and perturbation ratios can only be found when the sensor noise is considered.

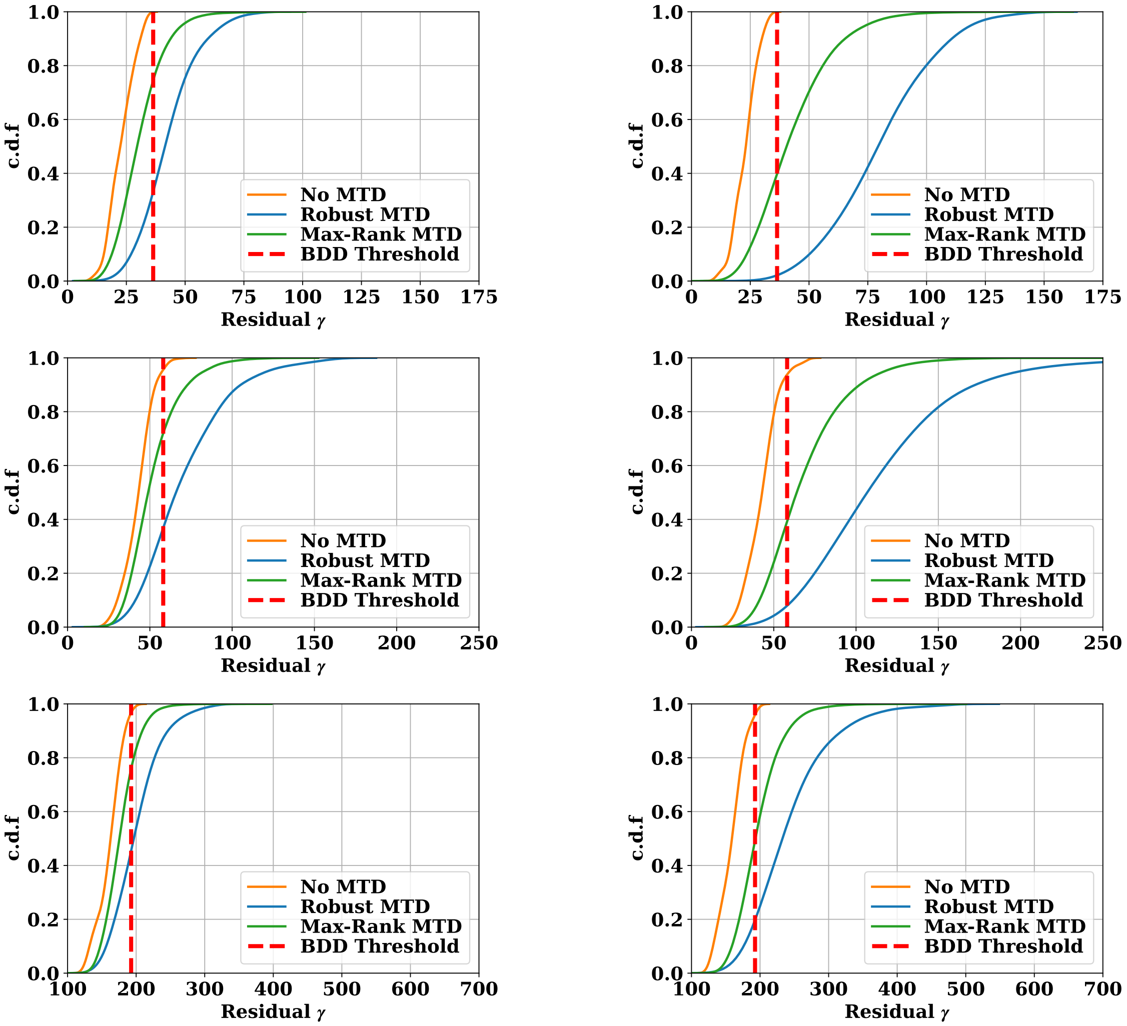


Figure 2.7: Residual distributions of AC-FDI attacks. The first row: case-6 system; the second row: case-14 system; the third row: case-57 system; the first column: attacks in range $[10, 15)$; the second column: attacks in range $[20, 25)$.

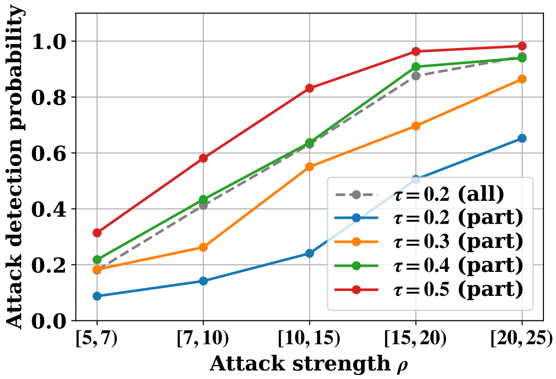


Figure 2.8: ADPs under different placements and perturbation ratios of D-FACTS devices.

Computational Time

The computational time of the proposed algorithms is summarized in Table 2.2. We test the proposed algorithm on the MacBook Pro with Apple M1 Pro chip and 32GB memory. For each system and algorithm, the computational times under all load conditions are recorded and averaged. The multi-run strategy is also applied to approach the global optimum of the nonlinear optimization problem which is also included in Table 2.2. Although the computation time depends on the system scales, number of D-FACTS devices, and algorithms, they are acceptable for real-time applications. In practice, as attackers spend time collecting new measurements and learning new parameters [73], the system operator can solve robust MTD algorithms with a period much longer than the state estimation time, e.g., several hours, or only change the Jacobian matrix $\boldsymbol{J}_N$ when the loads are significantly changed. A flat state vector may also be a choice to construct the Jacobian matrix if the loads change slowly.

Table 2.2: Computational Time (averaged by `no_load` runs).

Case	No. D-FACTS	Algorithm	Time (s)
case-6	11	(2.16)	0.022
case-14	20	Algorithm 2	1.925
	20	Algorithm 2 without (2.17)	0.325
	7	Algorithm 2	0.532
case-57	78	Algorithm 2	9.357

2.5 Conclusions

This chapter addresses the real-time robust implementation of MTD against unknown FDI attacks. The main contributions of this chapter are summarized in the following.

- This chapter proposes the concept of robust MTD in a noisy environment. It is theoretically proved that, for any given grid topology and MTD strategy, the minimal principal angle between the pre- and post-MTD Jacobian subspaces is directly linked with the worst-case performance against all potential FDI attacks, which can be used as a new criterion to represent the MTD effectiveness. Meanwhile, it is proved that the worst-case detection rate is proportional to the sine of this angle, with the impact of measurement noise being explicitly considered. This new criterion is an extension to the rank condition of the composite pre- and post-MTD matrix, which is only effective with noiseless measurements.

- A novel MTD design algorithm is formulated to improve the worst-case detection rate by maximizing the minimal principal angle under the complete grid configuration. Then it is demonstrated that the worst-case detection rate of the grid with incomplete configuration cannot be improved. Therefore, an iterative algorithm is formulated to maximize the minimal non-zero principal angle while limiting the chance of attacking on the subspace that cannot be detected.

- Numerical simulations on the IEEE 6-bus, 14-bus, and 57-bus systems demonstrate the improved detection performance of robust MTD algorithms against the worst-case, random, and single-state attacks, under both simplified and full AC models.

Chapter 3

Blending Data and Physics Against False Data Injection Attack: An Event-Triggered Moving Target Defence Approach

The robust MTD proposed in the previous chapter can effectively reduce the size of the attack space and maximize the detection rate of any unknown FDI attacks. However, the robust setting alters the transmission line reactance aggressively. Meanwhile, as conventional MTD is synchronized with state estimation, the conservative decision (to cover the entire attack space spatially and temporally) can inevitably cause significant operational cost. In addition, the black-box nature of the data-driven detector can cause uncontrollable FPR, which limits its wide application. Table 3.1 compares the MTD and the data-driven detector, showing a clear complement to each other. Although both data-driven and MTD are studied separately, few works have explored the overlap between these two areas. The authors in [58] apply data from the high-fidelity simulator to compensate for the inaccuracy of the detector based on an abstract grid model. Higgins, *et.al.* [59] proposed to trigger the MTD by the positive alarm of the data-driven detector. However,

Table 3.1: Comparison on MTD and data-driven detector

	Advantages	**Disadvantages**
MTD	High interpretability; Controllable FPR	High operation cost for frequent implementation
Data-Driven	Fast response; No extra operation cost	Low interpretability; Uncontrollable FPR

the work does not attempt to optimize the triggered MTD using the information from the data-driven detector.

Therefore, in the framework of sequential learning and model-based optimization, this chapter studies how a data-driven detector can be used to trigger the robust MTD and to adaptively reduce the size of uncertain attack space while improving its hiddenness to the attacker. In summary, this framework can be regarded as robust optimization with data-driven uncertainty set construction. In addition, the proposed sequential learning and optimization framework encodes prior knowledge on power system operation to demonstrate how physics can improve the detection accuracy and how MTD can help reject false positive decisions from the data-driven detector.

The remainder of the chapter is organized as follows. Preliminaries are given in Section 3.1. The proposed DDET-MTD algorithms are described in Section 3.2. The results are analyzed in Section 3.3 and this chapter concludes in Section 3.4.

3.1 Preliminaries

In this chapter, similar AC power flow models, the state estimation procedure, the definition of FDI attacks, and the implementation of MTD as section 2.1.2 are adopted. Any relevant refinement of the settings and the data-driven LSTM-AE detector are also explained. Similarly as before, vectors and matrices are represented by bold lower-case and upper-case letters, respectively. The zero and identity matrices are represented as $\boldsymbol{O}$ and $\boldsymbol{I}$ with appropriate dimensions. $[\cdot]$ represents the diagonalization on a vector; $(\cdot)^*$ and $(\cdot)^T$ represent the conjugate on a complex vector and matrix transpose. The Hadamard product is represented as $\circ$. The operators $\mathcal{P}(\boldsymbol{A}) = \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$ and $\mathcal{S}(\boldsymbol{A}) = \boldsymbol{I} - \mathcal{P}(\boldsymbol{A})$ represent the orthogonal projection matrices on $\text{Col}(\boldsymbol{A})$ and $\text{Ker}(\boldsymbol{A}^T)$, while $\mathcal{P}_{\boldsymbol{W}}(\boldsymbol{A})$ and

$\mathcal{S}_{\boldsymbol{W}}(\boldsymbol{A})$ represent the same operations with weight matrix $\boldsymbol{W}$. The symbol of probability is denoted as $\mathbb{P}$.

3.1.1 System Model

Referring to the Section 2.1.2, the power flow equations can be summarized as

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e} \tag{3.1}$$

where $\boldsymbol{e}$ is the zero-mean Gaussian measurement error with diagonal covariance matrix $\boldsymbol{R} = \text{diag}[\sigma_1^2, \sigma_2^2, \dots, \sigma_m^2]$. In this chapter, the RTU measurements and the corresponding mathematical formulations are considered as follows [141]:

1). Complex power injections:

$$\boldsymbol{S}_{bus} = [\boldsymbol{v}]\boldsymbol{Y}_{bus}\boldsymbol{v}^*$$

2). ‘from’ and ‘to’-side complex power flows:

$$\boldsymbol{S}_f = [\boldsymbol{C}_f\boldsymbol{v}]\boldsymbol{Y}_f^*\boldsymbol{v}^*$$

$$\boldsymbol{S}_t = [\boldsymbol{C}_t\boldsymbol{v}]\boldsymbol{Y}_t^*\boldsymbol{v}^*$$

where $\boldsymbol{v} \in \mathbb{C}^{N+1}$ is vector of complex bus voltages; $\boldsymbol{C}_f$ and $\boldsymbol{C}_t \in \mathbb{R}^{M\times(N+1)}$ are the ‘from’ and ‘to’ side incidence matrices, respectively; $\boldsymbol{Y}_{bus} \in \mathbb{R}^{(N+1)\times(N+1)}$ is the bus admittance matrix; $\boldsymbol{Y}_f$ and $\boldsymbol{Y}_t \in \mathbb{R}^{M\times(N+1)}$ are the ‘from’ and ‘to’ side branch incidence matrices, respectively. The total measurement becomes $\boldsymbol{z} = [\boldsymbol{P}_{bus}^T, \boldsymbol{P}_f^T, \boldsymbol{P}_t^T, \boldsymbol{Q}_{bus}^T, \boldsymbol{Q}_f^T, \boldsymbol{Q}_t^T]^T \in \mathbb{R}^{2N+4M+2}$. Detailed formulations of $\boldsymbol{Y}_{bus}$, $\boldsymbol{Y}_f$, and $\boldsymbol{Y}_t$ can be found in [141].

Based on the estimated state using SE in (2.2), the BDD raises an alarm if the measurement residual is greater than a predefined threshold [38]. In detail, letting $f_\chi(\gamma|\kappa)$ represent the density function of residual γ with DoF κ, the SO can decide the detection threshold $\tau_{\chi}(\alpha)$ by a tolerable FPR α such that

$$\mathbb{P}(\gamma \geq \alpha) = \int_{\tau_{\chi}(\alpha)}^{\infty} f_{\chi}(\gamma|\kappa)du = \alpha \tag{3.3}$$

3.1.2 FDI Attacks

Given the measurement $\boldsymbol{z}$, an attacker can launch FDI attacks by formulating $\boldsymbol{z}_a = \boldsymbol{z} + \boldsymbol{a}$, $\boldsymbol{a} \in \mathbb{R}^P$, which cannot be detected by the BDD if $\boldsymbol{a} = \boldsymbol{h}(\widehat{\boldsymbol{v}} + \boldsymbol{c}) - \boldsymbol{h}(\widehat{\boldsymbol{v}})$ [42]. Under this condition, $\boldsymbol{z}_a = \boldsymbol{h}(\boldsymbol{v} + \boldsymbol{c}) + \boldsymbol{e}$ whose residual is unchanged as (3.3). To successfully launch FDI attacks, the attacker's abilities are assumed as follows:

Assumption One. The attackers are aware of the topology and parameters of the grid to build $\boldsymbol{h}(\cdot)$, which can be circumvented by data-driven algorithms [142]. However, data collection is time-consuming, e.g. several hours [73].

Assumption Two. The attacker can access and modify all sensor measurements. This can be achieved by hijacking all RTU measurements or by changing the domain name system server between the SCADA front end and the control center [143]. Meanwhile, the attack strength is limited because the attacked state should be within the normal range [35].

Assumption Three. The attacker can verify his knowledge about the grid parameters by checking the integrity of the hijacked measurement. Similarly to BDD, the attacker can perform SE, and if the residual is greater than the threshold, the attacker will not carry out the attack but will turn to collecting more information [75].

Assumption One to Three require the attacker's effort to gain accurate gird topology and parameters, which may not be practical in real-time operation. However, we assume the strongest attacker and study the general defence algorithm against the unpredictable attacker, which is in line with the assumptions made in [68–70, 72, 73].

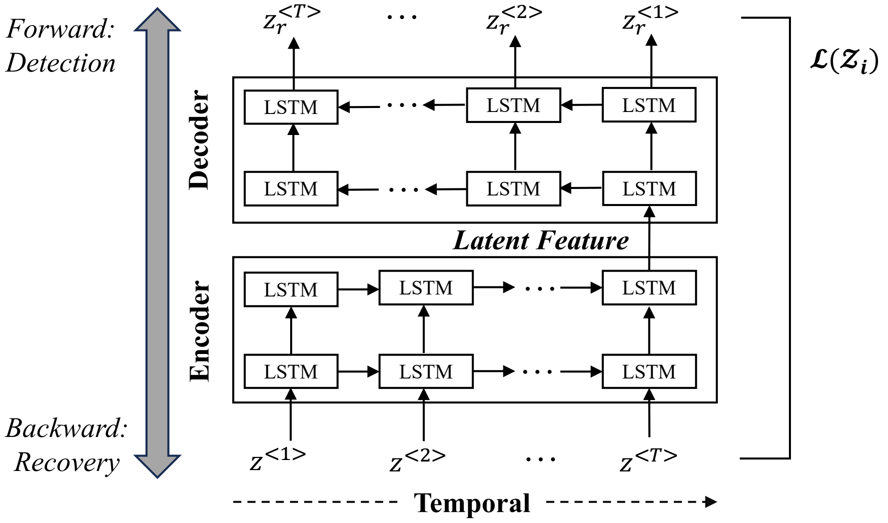


Figure 3.1: An illustrative structure of LSTM-AE neural network with four LSTM layers. In the forward pass, the measurements are reconstructed at the output with purpose of detecting attack, while in the backward recovery, the measurements are iteratively projected onto the normality manifold through gradient descent. Note that the backward here should be distinguished to the backward pass in automatic differentiation.

3.1.3 LSTM-AE based Data-Driven Detector

Although the attacker can launch FDI attacks by exploring the grid topology and parameters, the attacked measurement $\boldsymbol{z}_a^{<t>}$ at time t may violate the trend in a certain time window of length T. Our previous work in [131] designed a *semi-supervised* data-driven detector using LSTM-AE to explicitly learn the *spatio-temporal* correlations in sequential measurements. Fig. 3.1 illustrates the structure of LSTM-AE where each column of connected LSTM cells represents one layer of the deep recurrent network. Given a set of L normal measurements $\mathcal{Z} = \{\boldsymbol{z}^{<1>}, \boldsymbol{z}^{<2>}, \dots, \boldsymbol{z}^{<L>}\}$, consider a length-$T$ continuous subset $\mathcal{Z}_i = \{\boldsymbol{z}^{<t_i>}, \boldsymbol{z}^{<t_i+1>}, \cdots, \boldsymbol{z}^{<t_i+T-1>}\}$. At each layer, LSTM cells contain 'states' whose values depend on the previous memories and can be updated or forgotten by the current measurement. To learn the temporal pattern of the measurements, the LSTM-AE is trained to compress its input $\mathcal{Z}_i$ into a latent representation of the lower dimension, while only normal data can be successfully recovered by the decoder; thus, real-time attack measurements can be distinguished by directly evaluating the loss function:

$$\mathcal{L}(\mathcal{Z}_i) = \frac{1}{TP}\sum_{j=0}^{T-1} \left\|\boldsymbol{z}^{<t_i+j>} - f_d\left(f_e\left(\boldsymbol{z}^{<t_i+j>}\right)\right)\right\|_2^2 \tag{3.4}$$

where f_e and f_d represent the encoder and decoder mappings respectively. The detection threshold τ_{lstm} can be defined based on the distribution of the residual $\mathcal{L}(\mathcal{Z}_i)$ in the validation set [131]. Although the attacker may also exploit the temporal correlations between the measurements, we assume that they cannot know the exact temporal pattern learned from the LSTM-AE detector.

3.1.4 Moving Target Defence

Compared to the data-driven detector, model-based detections are more likely accepted by the system operator due to its high interpretability. To overcome the static nature of the model-based detector, MTD is introduced to proactively change the grid parameters using D-FACTS devices. The typical reactance perturbation ratio is less than 50% [73]. For convenience, the constraint on the reactance is converted to the constraint on the susceptance as follows.

$$\boldsymbol{h}_{\boldsymbol{b}_0}(\cdot) \xrightarrow{\text{MTD}} \boldsymbol{h}_{\boldsymbol{b}'}(\cdot)$$

where $\boldsymbol{b}' = \boldsymbol{b}_0 + \Delta\boldsymbol{b}$ are the susceptances after activating the D-FACTS devices. Details on the reactance to susceptance conversion can be found in Appendix D.1. Physical constraints can be represented by the set $\mathcal{B} = \{\boldsymbol{b}'|\boldsymbol{b}^- \leq \boldsymbol{b}' \leq \boldsymbol{b}^+\}$ where $\boldsymbol{b}^-$ and $\boldsymbol{b}^+$ are the lower and upper bound of the susceptance. If there is no D-FACTS device in branch i, $\boldsymbol{b}_i^- = \boldsymbol{b}_i^+ = \boldsymbol{b}_{0i}$.

If there is no attack, the post-MTD measurement still follows the χ^2 distribution. Therefore, MTD does not introduce additional FPR. On the contrary, if the attack exists, the residual vector will no longer follow the χ^2 distribution of the legitimate measurement and hence trigger the BDD alarm. In detail, *MTD effectiveness* refers to the accuracy of BDD after MTD is activated [68]. Recent literature also proposes the concept of *MTD*

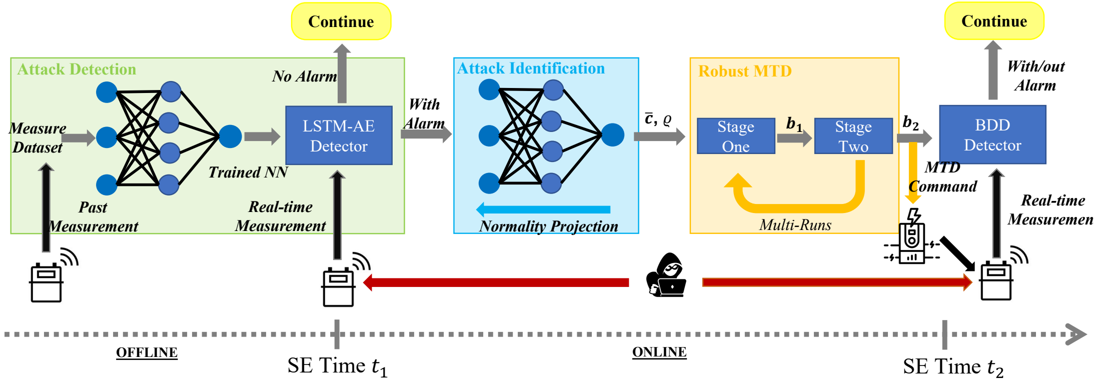


Figure 3.2: The DDET-MTD framework in one execution cycle. An LSTM-AE neural network is trained offline on normal measurement dataset. The same NN is used for online attack detection and identification. If an alarm is raised, the identified attack uncertainty set is fed into the two-stage robust MTD optimization to accept or reject the alarm from the data-driven detector.

hiddenness by noticing that the prudent attacker can also check the integrity of model parameters using the BDD-like method [75]. According to Assumption Three, the system will therefore face new threats [75]. Apart from achieving high detection rate, the hidden MTD requires reducing the attacker's residual so that the attackers keep using out-of-date grid knowledge to formulate the attack.

3.2 Data Driven Event-Triggered MTD

As shown in Fig. 3.2, the proposed DDET-MTD has three successive components in one execution cycle. First, the LSTM-AE detector in Section 3.1.3 is trained on the normal dataset offline and then tests the sensor measurement collected from SCADA in real-time operation. If a positive alarm is raised at SE time t_1, an attack identification algorithm is implemented to approximately extract the attack vector in the second component using the same neural network. The attack identification serves as the bridge between the data and physics by applying the extracted attack knowledge to the MTD design, in the meantime, reduces the execution cost of MTD and improves its hiddenness. In the last component, based on the identified attack, a robust MTD algorithm is triggered to verify the positive alarm from the LSTM-AE detector at the next SE time t_2. Intuitively, the false alarms from the data-driven detector can be sufficiently rejected by the subsequent MTD due to the controllable FPR of the MTD.

3.2.1 Physics-Informed Attack Identification

The LSTM-AE detector defines a manifold for normal measurement. Therefore, the attack identification can be achieved by first recovering the normal measurement toward the manifold of the LSTM-AE detector. Following Section 3.1.3, given a continuous measurement set $\mathcal{Z}_i = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_T\}$ with positive alarm by the LSTM-AE detector, we assume that only the last measurement vector is anomalous. Let the anomalous attack and recovered measurement be $\boldsymbol{z}_a$ and $\boldsymbol{z}_T^{nor}$, respectively. To explicitly encode the measurement equation (3.1), the recovered measurement can be written as

$$\boldsymbol{z}_T^{nor} = \boldsymbol{h}(\boldsymbol{v}_{R,T}^{nor}, \boldsymbol{v}_{I,T}^{nor}) \tag{3.5}$$

where $\boldsymbol{v}_{R,T}^{nor}$ and $\boldsymbol{v}_{I,T}^{nor}$ are the recovered real and imaginary voltage vectors. Here, the rectangular form on complex number is used to ensure stable back-propagation in Neural Network. Let $\mathcal{Z}_i^{nor} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_T^{nor}\}$. An energy function measuring the distance from $\mathcal{Z}_i^{nor}$ to the normality manifold defined by the LSTM-AE can be written as:

$$\mathcal{E}(\boldsymbol{v}_{R,T}^{nor}, \boldsymbol{v}_{I,T}^{nor}) = \mathcal{L}(\mathcal{Z}_i^{nor}) + \beta_R \|\boldsymbol{v}_{R,T}^{nor} - \boldsymbol{v}_{R,a}^{nor}\|_1 + \beta_I \|\boldsymbol{v}_{I,T}^{nor} - \boldsymbol{v}_{I,a}^{nor}\|_1 \tag{3.6}$$

Eq. (3.6) can be viewed as a non-linear Lasso regression on decision variable $(\boldsymbol{v}_{R,T}^{nor}, \boldsymbol{v}_{I,T}^{nor})$ where the projection of the attack measurement $\boldsymbol{z}_a$ on the LSTM-AE manifold is calculated with physical information (3.5) considered. In detail, the first term in (3.6) is the reconstruction loss (3.4) of the recovered normal measurement $\boldsymbol{z}_T^{nor}$, while the second and third terms penalize the difference between real and imaginary-part voltage deviations with weights β_R and β_I, respectively. Since the attack is usually sparse, the l_1-norm is used to regularize the number of attacked states. The l_1-norm is also less sensitive to the attack vector than the l_2-norm used in $\mathcal{L}(\mathcal{Z}_i^{nor})$ of (3.4).

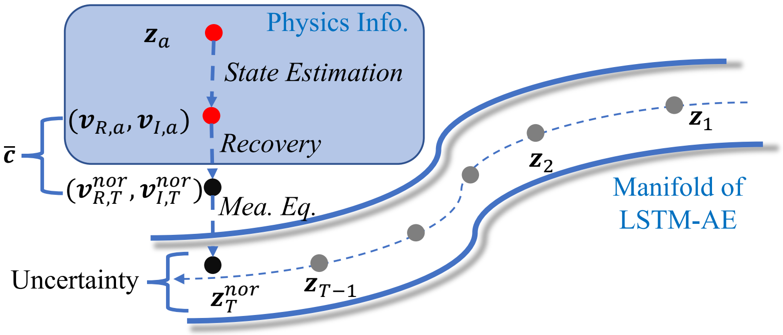


Figure 3.3: Illustration on attack identification algorithm.

The attack identification algorithm is illustrated in Fig. 3.3 and Algorithm 3. As shown in Fig. 3.3, the physics information is encoded through the measurement equation (3.1) and SE (2.2) when projecting the attack measurement onto the normal manifold defined by the LSTM-AE detector. Therefore, the main component of Algorithm 3 is to recover $\boldsymbol{z}_T^{nor}$ seen by both the BDD and the LSTM-AE detector. In line 3, the state estimation of the previous measurement is used as the warm start. `Adam` Optimizer [144] is used to minimize weighted loss $\mathcal{E}^k$ (3.6) by backpropagation with step size $lr_{\text{identifier}}$. The iteration in lines 5-14 is terminated if the reconstruction loss (3.4) is lower than the threshold τ_{lstm} or the maximum iteration number ite_{max} is achieved. The minimum iteration number ite_{min} is designed for warm-up purposes. Finally, the attack vector is identified by subtraction in line 15.

Given the i-th attack in an attack index set $\mathcal{I}_a$, the attack identification uncertainty set can be empirically determined as $\mathcal{C}_i = \{\boldsymbol{c}' | \|\boldsymbol{c}' - \bar{\boldsymbol{c}}_i\|_2^2 < \varrho^2\}$ where ϱ is the empirical upper bound on the deviation between the identified attack vector $\bar{\boldsymbol{c}}_i$ and the ground truth $\boldsymbol{c}_i$ for any $\forall i \in \mathcal{I}_a$.

In summary, Algorithm 3 guarantees that the recovered measurement can bypass the LSTM-AE detector and BDD. Therefore, the recovered state obeys the physics rules of power system. It also takes advantage of the formulation on FDI attacks so that the identified attack vector can lead to a stealthy attack, further improving the identification accuracy.

Algorithm 3: Attack Identification

```
Input  : $\mathcal{L}(\cdot)$, $\mathcal{Z}_i$, $lr_{\text{identifier}}$, $\beta_R$, $\beta_I$, $\tau_{\text{lstm}}$, $ite_{\min}$, $ite_{\max}$
Output: Identified attack vector $\bar{\boldsymbol{c}} = (\bar{\boldsymbol{c}}_R, \bar{\boldsymbol{c}}_I)$
Do state estimation on $\boldsymbol{z}_T^a$ as $\boldsymbol{v}_{R,T}^a$ and $\boldsymbol{v}_{I,T}^a$, and on $\boldsymbol{z}_{T-1}$ as $\boldsymbol{v}_{R,T-1}$ and $\boldsymbol{v}_{I,T-1}$
$k = 1$
$\boldsymbol{v}_{R,T}^k = \boldsymbol{v}_{R,T-1}$, $\boldsymbol{v}_{I,T}^k = \boldsymbol{v}_{I,T-1}$ /* warm start */
Initialize Adam optimiser with $lr_{\text{identifier}}$
while $k \leq ite_{max}$ do
    $\boldsymbol{z}_T^k = \boldsymbol{h}(\boldsymbol{v}_{R,T}^k, \boldsymbol{v}_{I,T}^k)$
    $\mathcal{Z}^k = \text{combine}\{\mathcal{Z}_i[1:T-1], \boldsymbol{z}_T^k\}$
    $\mathcal{E}^k = \mathcal{L}(\mathcal{Z}_i^k) + \beta_R \cdot \|\boldsymbol{v}_{R,T}^k - \boldsymbol{v}_{R,T}^a\|_1 + \beta_I \cdot \|\boldsymbol{v}_{I,T}^k - \boldsymbol{v}_{I,T}^a\|_1$
    if $k \geq ite_{min}$ or $\mathcal{L}(\mathcal{Z}_i^k) < \tau_{lstm}$ then
        break
    end
    $(\boldsymbol{v}_{R,T}^{k+1}, \boldsymbol{v}_{I,T}^{k+1}) \xleftarrow{\texttt{Adam}} \arg\min \mathcal{E}^k$
    $k \leftarrow k + 1$
end
$\bar{\boldsymbol{c}}_R = \boldsymbol{v}_{R,T}^a - \boldsymbol{v}_{R,T}^k$, $\bar{\boldsymbol{c}}_I = \boldsymbol{v}_{I,T}^a - \boldsymbol{v}_{I,T}^k$
```

3.2.2 Hidden and Effective MTD Algorithm

In the third component of DDET-MTD, the positive alarm of the LSTM-AE detector and the identified state attack vector can be used to trigger and design the MTD algorithm. Before introducing the idea of event triggering, the MTD algorithm is formulated as follows.

$$\min_{\boldsymbol{b}' \in \mathcal{B}} \quad \mathbb{P}(\text{Attacker can detect the MTD}) \tag{3.7a}$$

$$\text{subject to} \quad \mathbb{P}(\text{Operator can detect the attack}) \geq \rho \tag{3.7b}$$

Although the hiddenness of MTD is essential to deceive the prudent attacker, we argue that the main target of MTD is to detect the ongoing attack with high detection rate. Therefore, (3.7) is designed to minimize the attacker's chances of noticing the existence of MTD, subject to a specific detection accuracy ρ on the attack. However, this optimization problem is intrinsically hard to solve for two reasons. First, both the cost and the constraint in (3.7) are probabilistic and nonconvex, so the convergence property and global optimality are difficult to guarantee. Second, to guarantee detection accuracy, it requires the exact knowledge of the attack vector, which cannot be known in advance. To address the first problem, local linearizations are introduced in the measurement equation (3.1) on which convex relaxation is applied. For the second, a robust two-stage optimization problem is established based on the set of identification uncertainty set $\mathcal{C}_i$ from Algorithm 3.

Approximations of MTD Hiddenness

By explicitly considering the influence of susceptance on the measurement, the measurement equation (3.1) can be rewritten as $\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{v}, \boldsymbol{b}) + \boldsymbol{e}$. Under normal operation (no attack and MTD), the last iteration of SE is to solve the following normal equation:

$$\boldsymbol{z} - \boldsymbol{h}(\widehat{\boldsymbol{v}}, \boldsymbol{b}_0) = \boldsymbol{H}_{\widehat{\boldsymbol{v}}}(\boldsymbol{v} - \widehat{\boldsymbol{v}}) + \boldsymbol{e} \tag{3.8}$$

where $\boldsymbol{H}_{\widehat{\boldsymbol{v}}} = \left[\frac{\partial \boldsymbol{h}(\boldsymbol{v},\boldsymbol{b}_0)}{\partial \boldsymbol{v}}\right]_{\boldsymbol{v}=\widehat{\boldsymbol{v}}}$ and $\widehat{\boldsymbol{v}}$ is the state estimated before MTD. For the above system, the residual can be derived as

$$\gamma(\boldsymbol{z}, \boldsymbol{b}_0) = \|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}_{\widehat{\boldsymbol{v}}}\boldsymbol{e}\|_2^2 \tag{3.9}$$

where $\boldsymbol{S}_{\widehat{\boldsymbol{v}}} = \mathcal{S}_{\boldsymbol{R}^{-1}}(\boldsymbol{H}_{\widehat{\boldsymbol{v}}})$. $\gamma(\boldsymbol{z}, \boldsymbol{b}_0)$ also follows the χ^2 distribution with DoF $P - 2N$.

When the MTD is triggered, both $\boldsymbol{b}$ and $\boldsymbol{v}$ will be deviated from the stationary point. The first-order Taylor expansion around $(\widehat{\boldsymbol{v}}, \boldsymbol{b}_0)$ is written as:

$$\boldsymbol{z}' - \boldsymbol{h}(\widehat{\boldsymbol{v}}, \boldsymbol{b}_0) = \boldsymbol{H}_{\widehat{\boldsymbol{v}}}(\boldsymbol{v}' - \widehat{\boldsymbol{v}}) + \boldsymbol{H}_{\boldsymbol{b}_0}(\boldsymbol{b}' - \boldsymbol{b}_0) + \boldsymbol{e} \tag{3.10}$$

where $\boldsymbol{H}_{\boldsymbol{b}_0} = \left[\frac{\partial \boldsymbol{h}(\widehat{\boldsymbol{v}},\boldsymbol{b})}{\partial \boldsymbol{b}_0}\right]_{\boldsymbol{b}=\boldsymbol{b}_0}$.

Combining (3.8)-(3.10), the attacker's residual on the post-MTD measurement $\Delta\boldsymbol{z}'$ becomes:

$$\begin{aligned}\gamma(\boldsymbol{z}', \boldsymbol{b}_0) &= \|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}_{\widehat{\boldsymbol{v}}}(\boldsymbol{H}_{\widehat{\boldsymbol{v}}}(\boldsymbol{v}' - \widehat{\boldsymbol{v}}) + \boldsymbol{H}_{\boldsymbol{b}_0}(\boldsymbol{b}' - \boldsymbol{b}_0) + \boldsymbol{e})\|_2^2 \\ &= \|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}_{\widehat{\boldsymbol{v}}}(\boldsymbol{H}_{\boldsymbol{b}_0}(\boldsymbol{b}' - \boldsymbol{b}_0) + \boldsymbol{e})\|_2^2\end{aligned}$$

in which the second equality is due to the fact that $\boldsymbol{S}_{\hat{\boldsymbol{v}}}\boldsymbol{H}_{\hat{\boldsymbol{v}}} = 0$. Meanwhile, $\gamma(\boldsymbol{z}', \boldsymbol{b}_0)$ follows the non-central χ^2 distribution (NCX) with non-centrality parameter:

$$\lambda(\boldsymbol{z}', \boldsymbol{b}_0) = \|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}_{\widehat{\boldsymbol{v}}}\boldsymbol{H}_{\boldsymbol{b}_0}(\boldsymbol{b}' - \boldsymbol{b}_0)\|_2^2 \tag{3.11}$$

Since the probability that the MTD is detected by the attacker increases monotonically as $\lambda(\boldsymbol{z}', \boldsymbol{b}_0)$ increases [74], $\lambda(\boldsymbol{z}', \boldsymbol{b}_0)$ should be minimized. This result is consistent with the findings in [72, 75, 77] where the measurement change before and after MTD should be small. Note that both $\boldsymbol{S}_{\widehat{\boldsymbol{v}}}$ and $\boldsymbol{H}_{\boldsymbol{b}_0}$ are constants for a given load condition. Meanwhile, $\boldsymbol{H}_{\boldsymbol{b}_0}$ can be derived analytically using methods similar to those in [145].

Approximation of MTD Effectiveness

To accelerate the convergence speed and performance of SE, dishonest SE is widely used, in which the Jacobian matrix remains unchanged throughout the iteration [69]. The last iteration of dishonest SE on $\boldsymbol{z}'$ is represented as:

$$\boldsymbol{z}' - \boldsymbol{h}'(\widehat{\boldsymbol{v}}') = \boldsymbol{H}'_{\boldsymbol{v}_0}(\boldsymbol{v}' - \widehat{\boldsymbol{v}}') + \boldsymbol{e} \tag{3.12}$$

where $\widehat{\boldsymbol{v}}'$ is the estimated state of $\boldsymbol{z}'$ and $\boldsymbol{H}'_{\boldsymbol{v}_0} = \left[\frac{\partial \boldsymbol{h}'(\boldsymbol{v})}{\partial \boldsymbol{v}}\right]_{\boldsymbol{v}=\boldsymbol{v}_0}$.

The residual of the above system is derived as $\gamma(\boldsymbol{z}', \boldsymbol{b}') = \|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}'_{\boldsymbol{v}_0}\boldsymbol{e}\|_2^2$ where $\boldsymbol{S}'_{\boldsymbol{v}_0} = \mathcal{S}_{\boldsymbol{R}^{-1}}(\boldsymbol{H}'_{\boldsymbol{v}_0})$. Similarly, $\gamma(\boldsymbol{z}', \boldsymbol{b}')$ follows the χ^2 distribution with DoF $P - 2N$.

When an attack exists, $\boldsymbol{a} = \boldsymbol{h}(\widehat{\boldsymbol{v}}'_a + \boldsymbol{c}) - \boldsymbol{h}(\widehat{\boldsymbol{v}}'_a)$ where $\widehat{\boldsymbol{v}}'_a$ is the estimated state from the attacker after the MTD is triggered. As required by the MTD hiddenness, the difference in pre- and post-MTD measurements is minimized. Therefore, it is reasonable to assume that $\widehat{\boldsymbol{v}}'_a$ is close to $\widehat{\boldsymbol{v}}$. Following Assumption Two, for small state injection, the attack vector can be approximated as $\boldsymbol{a} = \boldsymbol{H}_{\widehat{\boldsymbol{v}}}\boldsymbol{c}$ [80]. Consequently, the non-centrality parameter of the post-MTD measurement under attack is approximated as:

$$\lambda(\boldsymbol{z}'_a, \boldsymbol{b}') = \|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}'_{\boldsymbol{v}_0}\boldsymbol{H}_{\widehat{\boldsymbol{v}}}\boldsymbol{c}\|_2^2 \tag{3.13}$$

Attack-Aware Robust MTD Reformulation

Based on the approximations of the hiddenness (3.11) and effectiveness (3.13) of MTD, the probabilistic optimisation problem (3.7) becomes nonprobabilistic for a given attack $\boldsymbol{c}$:

$$\min_{\boldsymbol{b}'\in\mathcal{B}} \quad \lambda(\boldsymbol{z}', \boldsymbol{b}_0) \tag{3.14a}$$

$$\text{subject to} \quad \lambda(\boldsymbol{z}'_a, \boldsymbol{b}') \geq \lambda_c(\rho) \tag{3.14b}$$

In (3.14a), only when $\lambda(\boldsymbol{z}', \boldsymbol{b}_0) = 0$, the MTD can be 100% hidden to the attacker. In most cases, the hiddenness and effectiveness of MTD have been shown to be contradictory [75–77]. In (3.14b), the probability constraint (3.7b) is converted non-probabilistic. In fact, there is a $\lambda_c(\rho)$ such that the detection rate at $\boldsymbol{c}$ is equal to ρ [74]:

$$\mathbb{P}(\gamma \geq \tau(\alpha)) = \int_{\tau(\alpha)}^{\infty} f_{\chi}(\gamma|\kappa, \lambda_c) = \rho \tag{3.15}$$

where $f_{\chi}(\gamma|\kappa, \lambda_c)$ represents the density function of the NCX distribution with DoF $\kappa = P - 2N$ and the non-centrality parameter equals λ_c.

Optimization (3.14) still requires exact knowledge of the attack vector $\boldsymbol{c}$, which is not available for the operator. Therefore, a robust reformulation of (3.14) is derived by guaranteeing the lowest detection rate for the attacks in the attack uncertainty set $\mathcal{C}$ defined in Section 3.2.1:

$$\min_{\boldsymbol{b}'} \quad \lambda(\boldsymbol{z}', \boldsymbol{b}_0) \tag{3.16a}$$

$$\text{subject to} \quad \boldsymbol{b}' \in \mathcal{B} \tag{3.16b}$$

$$\min_{\boldsymbol{c}'\in\mathcal{C}} \lambda(\boldsymbol{z}'_a, \boldsymbol{b}') \geq \lambda_c(\rho) \tag{3.16c}$$

Problem (3.16) is a bilevel optimization problem [146]. The objective of the upper level is to decrease the chance that the attacker detects MTD. The decision variable in upper level is the MTD setpoint $\boldsymbol{b}'$ and the constraint on $\boldsymbol{b}'$ is the permissible set of D-FACTS devices $\mathcal{B}$. At the lower level, the objective function is to find the state injection that results in the lowest detection rate, subject to the set of uncertainties $\mathcal{C}$. Note that the upper level decision variable $\boldsymbol{b}'$ is nested at the lower level parametrically. The nesting structure robustly ensures that all possible attacks in $\mathcal{C}$ can be detected with predefined probability ρ.

To simplify the analysis, only active power flow measurements are considered for MTD effectiveness, as active power is more important in state estimation and sensitive to changes in voltage phase angle [81]. As a result, the Jacobian matrix in (3.12) can be analytically written as:

$$\boldsymbol{H}_{\boldsymbol{v}_0} = \underbrace{\boldsymbol{V} \cdot \boldsymbol{G} \cdot \boldsymbol{A}_r^s}_{\boldsymbol{C}} - \boldsymbol{V} \cdot \boldsymbol{B} \cdot \boldsymbol{A}_r^c \tag{3.17}$$

where $\boldsymbol{V} = [(\boldsymbol{C}_f \boldsymbol{v}_0) \circ (\boldsymbol{C}_t \boldsymbol{v}_0)]$; $\boldsymbol{G} = [\boldsymbol{g}]$; $\boldsymbol{B} = [\boldsymbol{b}]$; $\boldsymbol{A}_r^s = [\sin \boldsymbol{A}\boldsymbol{\theta}_{\mathbf{0}}]\boldsymbol{A}_r$; and $\boldsymbol{A}_r^c = [1/\boldsymbol{t}][\cos \boldsymbol{A}\boldsymbol{\theta}_{\mathbf{0}}]\boldsymbol{A}_r$. $\boldsymbol{A}_r$ is the reduced incidence matrix by removing the column that represents the reference bus from the incidence matrix $\boldsymbol{A}$; $\boldsymbol{t}$ is the vector of the transformer tap ratio. The detection threshold corresponding to the active power flow measurements is denoted as λ_c'. Intuitively, guaranteeing the detection rate on a subset of the measurement can also guarantee the detection rate on the full measurement due to the increased redundancy.

As proved by [74], only when the attack strength is greater than a certain threshold can $\lambda(\boldsymbol{z}_a', \boldsymbol{b}') \geq \lambda_c'$ be satisfied. Therefore, despite the non-linearity and non-convexity, (3.16) may not have a feasible solution. As a result, (3.16) is separated into two stages. In stage one, the feasibility of constraint (3.16c) is checked by maximizing its left hand side. The optimal solution of stage one is then used as the feasible warm start in stage two to improve its hiddenness.

Convex Stage-One Optimization

In stage one, the feasibility of constraint (3.16c) is checked by maximizing the detection rate on the worst-case attack in $\mathcal{C}$

$$\max_{\boldsymbol{b}'\in\mathcal{B}} \min_{\boldsymbol{c}'\in\mathcal{C}} \lambda(\boldsymbol{z}'_a, \boldsymbol{b}') \tag{3.18}$$

Multi-run strategy is required to solve the non-convex problem (3.18) with different starting points in $\mathcal{B}$. For each run, an equivalent convex reformulation is derived as follows:

Proposition 4. Define auxiliary variable $\omega \in \mathbb{R}$, $\nu \in \mathbb{R}$, $\boldsymbol{H}_1 = \boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{H}_{\widehat{\boldsymbol{v}}}$, and $\boldsymbol{H}'_0 = \boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{H}'_{\boldsymbol{v}_0}$. The problem (3.18) is equivalent to the following:

$$\begin{aligned}
\max_{\boldsymbol{b}',\nu,\omega} \quad & \omega && \text{(3.19a)}\\
\text{subject to} \quad & [\boldsymbol{b}'] - [\boldsymbol{b}^-] \succeq 0, [\boldsymbol{b}^+] - [\boldsymbol{b}'] \succeq 0 && \text{(3.19b)}\\
& \nu \geq 0 && \text{(3.19c)}\\
& \begin{bmatrix} \nu(\bar{\boldsymbol{c}}^T\bar{\boldsymbol{c}} - \varrho^2) - \omega & \nu\bar{\boldsymbol{c}}^T & \boldsymbol{O} \\ \star & \nu\boldsymbol{I} + \boldsymbol{H}_1^T\boldsymbol{H}_1 & \boldsymbol{H}_1^T\boldsymbol{H}'_0 \\ \star & \star & {\boldsymbol{H}'_0}^T\boldsymbol{H}'_0 \end{bmatrix} \succeq 0 && \text{(3.19d)}
\end{aligned}$$

The proof can be found in Appendix B.5.

Referring to (3.17), the only nonlinearity of (3.19) is in the last block-diagonal entry of (3.19d). To linearize ${\boldsymbol{H}'_0}^T\boldsymbol{H}'_0$, iterative algorithm is designed with starting point $\boldsymbol{b}_0$ and the following proposition is derived:

Proposition 5. *Let $\boldsymbol{C}^N = \boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{C}$ and $\boldsymbol{V}^N = \boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{V}$. Define $\boldsymbol{b}_k$ as the feasible solution of the k-th iteration. A sufficient convex condition for* (3.19d) *is*

$$\begin{bmatrix} \nu(\bar{\boldsymbol{c}}^T\bar{\boldsymbol{c}} - \varrho^2) - \omega & \nu\bar{\boldsymbol{c}}^T & \boldsymbol{O} \\ \star & \nu\boldsymbol{I} + \boldsymbol{H}_1^T\boldsymbol{H}_1 & \boldsymbol{H}_1^T\boldsymbol{H}'_0 \\ \star & \star & \boldsymbol{H}_{update} \end{bmatrix} \succeq 0 \tag{3.20}$$

with

$$\begin{aligned}\boldsymbol{H}_{update} = \quad & (\boldsymbol{V}^N[\boldsymbol{b}_k]\boldsymbol{A}_r^c)^T(\boldsymbol{C}^N + \boldsymbol{V}^N[\boldsymbol{b}']\boldsymbol{A}_r^c) + (\boldsymbol{C}^N + \boldsymbol{V}^N[\boldsymbol{b}']\boldsymbol{A}_r^c)^T(\boldsymbol{V}^N[\boldsymbol{b}_k]\boldsymbol{A}_r^c) \\ & -(\boldsymbol{V}^N[\boldsymbol{b}_k]\boldsymbol{A}_r^c)^T(\boldsymbol{V}^N[\boldsymbol{b}_k]\boldsymbol{A}_r^c)\end{aligned}$$

The proof can be found in Appendix B.6.

In summary, at the k-th iteration, the following convex programming is solved until convergence, though it may not converge to the global optimality of (3.18) and (3.19).

$$\begin{aligned}\max_{\boldsymbol{b}',\mu,\omega} \quad & \omega \\ \text{subject to} \quad & (3.19\text{b}), (3.19\text{c}), (3.20)\end{aligned} \tag{3.21}$$

Convex Stage-Two Optimisation

The stage-one problem checks the feasibility of (3.16c). In detail, if the optimal solution ω^* of (3.19) (or similarly the final iteration of (3.21)) is greater than λ_c', the original bilevel problem (3.16) can be solved with the optimal point of (3.19) as a feasible warm start. Otherwise, the threshold in (3.16c) should be reduced to $\omega^\star$ to have a feasible solution. In either situation, denoting the threshold of constraint (3.16c) after stage one as ω, the following proposition gives a feasible and convex reformulation to (3.16) in which the MTD effectiveness is guaranteed to the level determined by stage one while the hiddenness is improved.

Proposition 6. With all variables and parameters defined as in Proposition 4, and let auxiliary variable $\phi \geq 0$, $\boldsymbol{H}_{\text{hid}} = \boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}_{\widehat{\boldsymbol{v}}}\boldsymbol{H}_b$. The bilevel optimisation problem (3.16) with $\lambda_c(\rho)$ replaced by ω can be solved by

$$\min_{\boldsymbol{b}',\nu,\phi} \quad \phi \tag{3.22a}$$

$$\text{subject to} \quad (3.19\text{b}), (3.19\text{c}), (3.19\text{d}) \tag{3.22b}$$

$$\begin{bmatrix} \phi & (\boldsymbol{b}' - \boldsymbol{b}_0)^T\boldsymbol{H}_{\text{hid}}^T \\ \star & \boldsymbol{I} \end{bmatrix} \succeq 0 \tag{3.22c}$$

The proof can be found in Appendix B.7.

Similarly, the non-convexity in ${\boldsymbol{H}'_0}^T \boldsymbol{H}'_0$ can be solved iteratively by the sufficient condition described in Proposition 5. This results in an iterative algorithm to solve the stage-two problem:

$$\begin{aligned} \max_{\boldsymbol{b}',\mu,\phi} \quad & \phi \\ \text{subjective to} \quad & (3.19\text{b}), (3.19\text{c}), (3.20), (3.22\text{c}) \end{aligned} \tag{3.23}$$

where ωs in (3.22) and (3.23) are constants determined by the optimum of stage one.

In summary, the two-stage optimization in DDET-MTD has been developed to guarantee the effectiveness of MTD while improving hiddenness. Based on convex relaxation, the hidden and effective MTD can be designed as follows:

1. Solve the stage-one problem (3.21) iteratively with different start point $\boldsymbol{b}_0$. Store the multi-run results of $\boldsymbol{b}'$ in a set $\mathcal{D}^{\text{one}}$ and the corresponding cost ω into a set $\mathcal{G}^{\text{one}}$.

2. If the largest cost in $\mathcal{G}^{\text{one}}$ is smaller than λ'_c, use the corresponding susceptance in $\mathcal{D}^{\text{one}}$ as a warm start in stage-two problem (3.23) and solve it iteratively.

3. If the largest cost in $\mathcal{G}^{\text{one}}$ is larger than or equal to λ'_c, define the index set $\mathcal{I}^{\text{two}} = \{i | \omega_i \geq \lambda'_c, \omega_i \in \mathcal{G}^{\text{one}}\}$ and candidate warm-start susceptance set $\mathcal{D}^{\text{two}} = \{\mathcal{D}[i], i \in \mathcal{I}^{\text{two}}\}$. For each $\boldsymbol{b} \in \mathcal{D}^{\text{two}}$, iteratively solve stage-two problem (3.23). The optimal susceptance is returned with the smallest cost.

The detailed algorithm can be found in Appendix D.2.

3.3 Simulations and Results

3.3.1 Simulation Settings

Model Configurations

The proposed DDET-MTD algorithm is tested on the IEEE case-14 system [141]. Although we have derived the theoretical analysis using simplified models, all the simulations are implemented under full AC condition. Real-time load consumptions and photovoltaic generations are assigned to each bus for four months using a similar method in [14]. The load data is interpolated to 5-min resolution, resulting in over 35k data in total. For each operation instance, AC-OPF is solved by `PyPower` [141]. The standard deviation of the measurement noise is set to 2% of the default values in the case-14 system case file. The FPR of BDD is set as $\alpha = 2\%$. The MTD threshold λ_c and λ_c' are determined by (3.15) with $\rho = 1 - \alpha = 98\%$. LSTM-AE attack detection and identification algorithms are trained and implemented using `PyTorch` [147] with hyperparameters summarised in Table 3.2. The data set is separated into 60% training, 20% validation, and 20% test sets. Throughout the simulation, random sparse AC-FDI attacks are generated with the number of attacked buses equal to 1-3, and the strength of the attacks is set as $\pm 10\% - 20\%$ and $\pm 20\% - 30\%$ of the normal state solved from the real-time measurements. For example, the pair $(2, 0.3)$ means that there are two buses being attacked with strength at random in $\pm 0.2 - 0.3$. In the simulation, 200 attacks are randomly generated from the entire test set for each type of attack. Without losing generality, all the branches are equipped with D-FACTS devices and the maximum reactance perturbation ratio is 50% [73]. Furthermore, the convex MTD optimization problems are solved by `CVXPY` [148] with `MOSEK` solver. Hyperparameters for stage-one and stage-two optimizations are summarized in Table 3.3.

Table 3.2: Hyperparameters for the Detector and Identifier.

Sample Length	6	**Encoder Size**	68-48-29-10
Epochs No.	1000	**Batch Size**	32
$\boldsymbol{lr}_{\textbf{detector}}$	0.001	$\boldsymbol{lr}_{\textbf{identifier}}$	0.005
Early Stop Patience	10	**Early Stop Diff.**	0
$\boldsymbol{\beta_R}, \boldsymbol{\beta_I}$	0.1	**Optimizer**	Adam
$\boldsymbol{ite}_{\textbf{max}}$	1000	$\boldsymbol{ite}_{\textbf{min}}$	50

Table 3.3: Hyperparameters for Stage-One/Two Optimizations.

Multi-Run No. no	15
Max. iteration No. $ite_{\text{one}}, ite_{\text{two}}$	100
Tolerance of stage one tol_{one}	0.1
Tolerance of stage two tol_{two}	1

Baseline Algorithms

Two algorithms, namely, the modified Max-Rank MTD [68–70] and (incomplete) Robust MTD [74], are implemented for benchmarking the proposed algorithm. In Max-Rank MTD, the D-FACTS devices are perturbed within $\mu_{\min}\boldsymbol{x}_i \leq |\Delta\boldsymbol{x}_i| \leq \mu_{\max}\boldsymbol{x}_i$ (with $\mu_{\max} > \mu_{\min} > 0$) so that the rank of the composite matrix is maximized, which results in maximum detection rate under noiseless assumption. Due to the randomness of this algorithm, we simulate 1000 attacks for each attack scenario under different load conditions and record the average performance. The Robust MTD algorithm considers maximizing the detection rate on the worst-case attack without any prior knowledge of the attack. Therefore, it can be viewed as a conservative formulation on DDET-MTD with an attack uncertainty set $\mathcal{C} = \mathbb{R}^N$. Although both baseline algorithms are periodic with SE, we also simulate their event-triggering variants, triggered by the same LSTM-AE.

Metrics

Four metrics are considered throughout the simulation.

From an attack defence perspective, Attack Detection Probability (ADP) and Defence Hiddenness Probability (DHP) can be used to assess the effectiveness and hiddenness of MTD as follows [77].

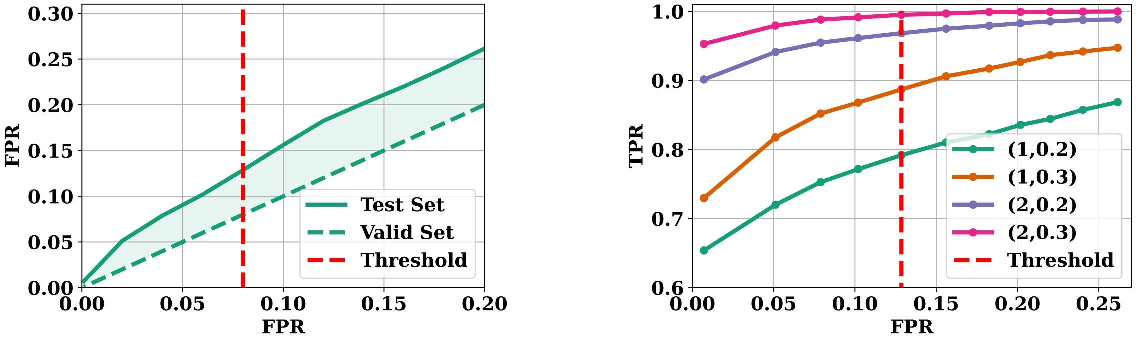
$$\text{ADP} = \frac{\text{Number of attacks being detected}}{\text{Total number of attacks}} \tag{3.24a}$$

$$\text{DHP} = \frac{\text{Number of MTDs not being detected}}{\text{Total number of MTDs}} \tag{3.24b}$$

From an economical perspective, the increase in average cost and the average perturbation ratio of reactance due to the trigger of the MTD are considered.

3.3.2 LSTM-AE Detector

Fig. 3.4 illustrates the TPR and FPR of the LSTM-AE detector. Various detection thresholds τ_{lstm}s are determined by the distribution of reconstruction losses in the valida-



(a)

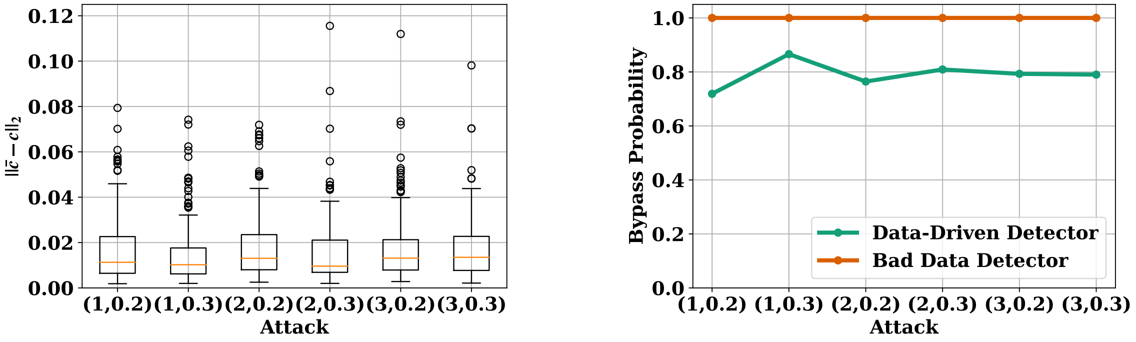
(b)

Figure 3.4: Performance of LSTM-AE attack detector: (a). Generalization error (FPR) on the test dataset; (b). ROC curves on different attacks.

tion set [131]. As shown in Fig. 3.4(a), the same detection threshold based on the FPR in the validation set can result in a higher FPR in the test set due to unseen load and PV patterns. The ROC curves on different types of attack are also summarized in Fig. 3.4(b), which clearly shows the trade-off between TPR and FPR. In detail, a larger attack results in a higher detection rate and to have 90% TPRs on all types of attack, at least 25% normal operation samples are incorrectly classified as attack. Since attack is rare in real-time operations, this high FPR can significantly influence normal operation. In the following simulation, τ_{lstm} corresponding to 8.0% FPR in the validation set is used as the detection threshold in the LSTM-AE detector, resulting in 12.84% FPR in the test set (highlighted by the red dotted line in Fig. 3.4).

3.3.3 LSTM-AE Identifier

Fig. 3.5 summarized the performance of the attack identification algorithm. As shown in Fig. 3.5(a), the average identification deviation is around 0.01p.u. and most of them are smaller than 0.04p.u.. As the average normal state angle in the simulation is 0.71p.u., the identification algorithm is accurate and stable under different attack scenarios. Fig. 3.5(b) tests whether the recovered measurement can bypass the BDD and LSTM-AE detector. First, since the identification algorithm filters the measurement noise by (3.5), the recovered measurement can certainly bypass the BDD. Second, due to the existence



(a)

(b)

Figure 3.5: Performance of LSTM-AE attack identification: (a). Identification deviation (in p.u.); (b). Probability of bypassing detectors.

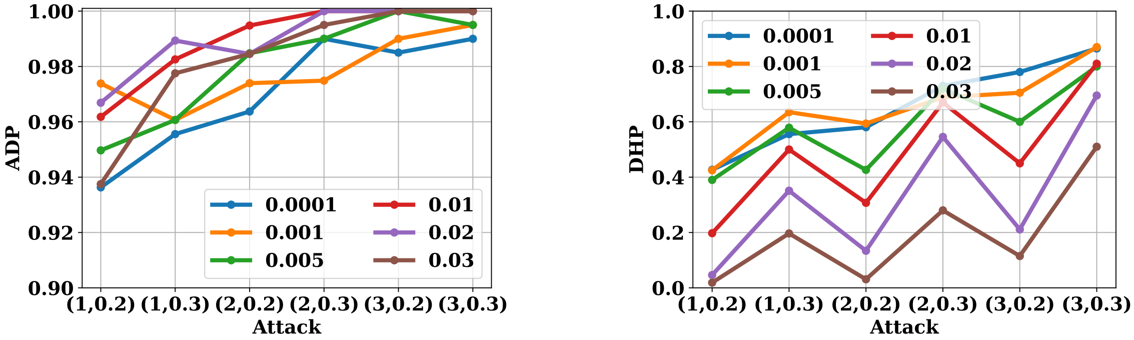
of regularization in the energy function (3.6) and the limit of iteration numbers, only 80% of the recovered measurement can bypass the LSTM-AE detector. Nonetheless, the reconstruction losses are much smaller than those of the attacked measurement, meaning that the recovered measurements are close to the normality manifold defined by the LSTM-AE detector. Therefore, the identified attack vector is quite accurate and can be used to guide the hidden and effective MTD algorithm.

3.3.4 Properties of DDET-MTD

In this section, we investigate the performance of the proposed DDET-MTD algorithm.

Sensitivity of ϱ

First, based on the identification accuracy in Fig. 3.5, the effectiveness and hiddenness of MTD are summarized in Fig. 3.6(a) and (b), respectively. In Fig. 3.6(a), a larger attack is more likely to be detected and, in general, ADP increases and then decreases slightly as ϱ increases. When ϱ is small, the MTD is optimized on the limited set of candidate attack vectors around the identified attack, which may not include the actual attack vector. On the contrary, when ϱ is large, the robust MTD is conservative by maximizing the detection rate on the worst possible attack in a larger set, causing the actual detection rate to decrease. An extreme example is that when $\varrho > \|\boldsymbol{c}\|_2$, a zero-state attack vector becomes



(a)

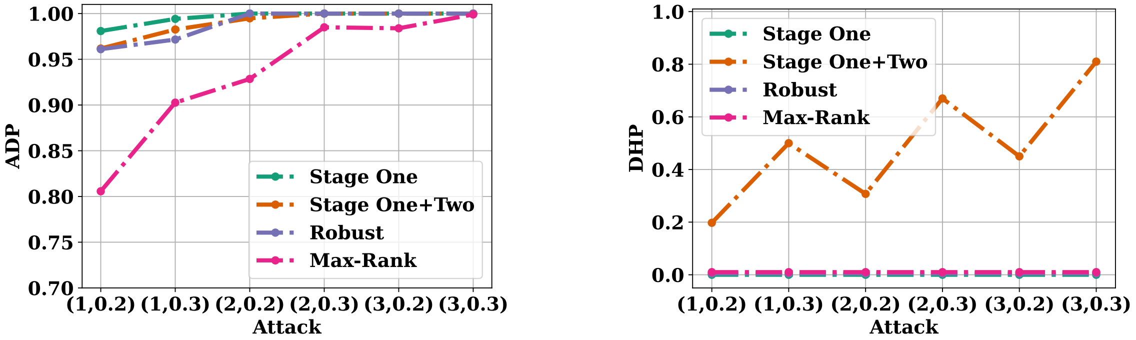
(b)

Figure 3.6: Evaluation on different ϱs: (a). MTD effectiveness; (b). MTD hiddenness.

the worst-case attack, leading to a trivial solution to (3.18). Regarding the hiddenness of MTD, Fig. 3.6(b) shows that MTDs on a strong attack result in high DHP, which implies the trade-off between hiddenness and effectiveness. Referring to (3.16), when the attack is strong, the effectiveness constraint can be more easily achieved, which in turn gives a lower residual of the attacker. Meanwhile, DHP decreases as ϱ increases, which can be explained by a similar reason. In the following, $\varrho = 0.01$ will be used as the uncertainty bound in $\mathcal{C}$ due to its high ADP and moderate DHP.

Comparison with Max-Rank and Robust MTDs

In this section, Stage-One and Stage-One + Stage-Two of the proposed DDET-MTD are compared with the Max-Rank MTD and Robust MTD algorithms. To fairly verify the performance of the proposed algorithm, both Max-Rank and Robust MTDs are triggered by the same LSTM-AE detector. Therefore, only the attacks that can be detected by the LSTM-AE are evaluated by the MTD algorithms, and we will leave a full comparison in the next section. First, as shown in Fig. 3.7(a), both Stage-One and Stage-One + Stage-Two can achieve ADPs greater than 96% in all attack cases. The ADP of Stage-One is slightly higher than that of Stage-One + Stage-Two when the attack strength is small. This is because Stage-One maximises the residual, while Stage-Two adds the threshold as constraint. The proposed algorithm has an ADP comparable to the Robust MTD, which is significantly higher than the Max-Rank MTD. Therefore, it can be concluded that the



(a) (b)

Figure 3.7: Comparison on Stage-One, Stage-One + Stage-Two in DDET-MTD, Max-Rank MTD, and Robust MTD, assuming that the attack is detected by the LSTM-AE detector: (a). MTD effectiveness; (b) MTD hiddenness.

event-triggered MTD does not significantly compromise the performance of the LSTM-AE detector shown in Fig. 3.4. Furthermore, Fig. 3.7(b) shows that adding Stage-Two can significantly improve MTD hiddenness without compromising ADP. On the contrary, without considering the hiddenness of MTD, Stage-One, Robust and Max-Rank MTDs can always be detected by the attacker once the detector raises an alarm.

3.3.5 Performances under Real-Time Operations

In this section, more realistic power system operation is considered. The proposed DDET-MTD is compared with the Max-Rank MTD and the Robust MTD in both **periodic** and **event-triggering** settings. Meanwhile, as cyberattacks are very rare in practice, it is reasonable to discuss MTD usage and generator cost without attacks to see how the extra defence can impact normal system operations. In addition, the false positive rate reduction of the LSTM-AE detector is also discussed.

Operations under FDI Attack

Average performances of Max-Rank MTD (Max), Robust MTD (Robust), and DDET-MTD (DDET) under different attack scenarios are compared in Table 3.4.

In general, DDET-MTD has the highest DHPs under each attack. Note that the DHPs of event-triggered Max-Rank and Robust MTDs are not zero due to the missing alarms

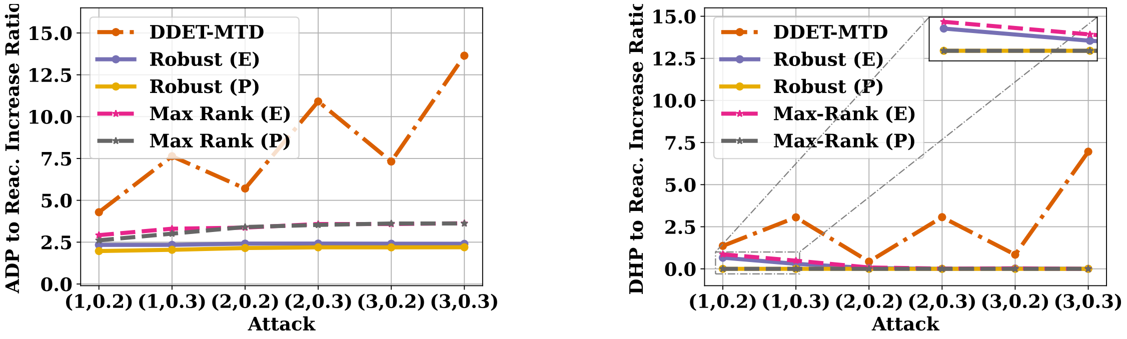
Table 3.4: Average MTD performance under different attacks (in %). The ↑ and ↓ represent the desired values being large and small, respectively. The best and second best performances are highlighted in red and blue, respectively.

Attack	Metric	Periodic		Event-Triggered		
		Max	Robust	Max	Robust	DDET
(1,0.2)	↑ ADP	71.90	90.00	65.10	74.00	75.50
	↑ DHP	0.00	0.00	19.20	23.00	41.24
	↓ Cost	0.17	0.64	0.14	0.47	0.02
	↓ Reac.	27.53	45.69	22.30	34.84	17.15
$(1, 0.3)$	↑ ADP	83.40	93.00	79.70	84.50	85.50
	↑ DHP	0.00	0.00	11.70	12.00	64.00
	↓ Cost	0.17	0.61	0.16	0.54	0.01
	↓ Reac.	27.73	45.50	24.12	40.21	11.01
$(2, 0.2)$	↑ ADP	93.10	98.50	90.81	98.00	95.50
	↑ DHP	0.00	0.00	2.20	2.00	34.72
	↓ Cost	0.17	0.60	0.16	0.60	0.01
	↓ Reac.	27.34	45.82	26.95	44.63	16.56
$(2, 0.3)$	↑ ADP	96.91	100.00	98.10	100.00	100.00
	↑ DHP	0.00	0.00	0.40	0.00	67.00
	↓ Cost	0.17	0.64	0.17	0.631	0.00
	↓ Reac.	27.45	45.63	23.38	45.54	9.12
$(3, 0.2)$	↑ ADP	98.90	99.00	97.61	99.00	100.00
	↑ DHP	0.00	0.00	0.80	1.00	45.00
	↓ Cost	0.18	0.62	0.17	0.64	0.00
	↓ Reac.	27.24	45.21	27.36	45.28	13.66
$(3, 0.3)$	↑ ADP	99.80	100.00	99.90	100.00	100.00
	↑ DHP	0.00	0.00	0.00	0.00	81.00
	↓ Cost	0.17	0.66	0.18	0.65	0.00
	↓ Reac.	27.53	45.54	27.57	45.85	7.33

(false negative samples) from the LSTM-AE detector. The false negative rate of the LSTM-AE detector also causes the lower ADP of DDET-MTD than the periodic Robust MTD when the attack strength is low (see Fig. 3.4(b)). However, periodic Robust MTD is the least economical method and cannot improve the hiddenness of MTD.

Thanks to the attack uncertainty set $\mathcal{C}$, the DDET-MTD can detect the attack with fewer efforts, resulting in the best economic performance of the lowest average reactance perturbation. Additionally, when the attack strength increases, the reactance perturbation ratio decreases, which can save the usage of D-FACTS devices in real-time operation. In contrast, as Robust MTD considers the worst detection performance all the time, it has the worst economic performance. The Robust and Max-Rank MTDs have almost constant average ratios per D-FACTS device under both periodic and event-triggering settings, as both algorithms cannot reflect different attack strengths and therefore can easily over-react most of the time.

Although optimization (3.7) does not take generator cost into account, simulation



(a)

(b)

Figure 3.8: a). Average ADP to average reactance increase ratio; (b). Average DHP to average reactance perturbation ratio. (E) and (P) represent the event-triggered and periodic settings, respectively.

shows that DDET-MTD results in the lowest cost increase under each attack for two reasons. First, the DDET-MTD has the minimum reactance deviation against the default reactance settings. Therefore, its operational point is closest to the optimal setting. Second, by improving the MTD hiddenness, the pre- and post- MTD power flows become similar to each other, resulting in less flow redistribution and similar line losses. Furthermore, Table 3.4 also illustrates that smaller costs are needed to detect more intense attacks in DDET-MTD, which is similar to the reactance perturbation.

To better illustrate the performance, Fig. 3.8 calculates the ratio of ADP and DHP with respect to the average perturbation ratio. It can be demonstrated that the DDET-MTD has the best trade-off between attack defence and operation economics, especially when the attack strength is high.

False Positive Rejection

Fig. 3.9(a) records the residuals of the LSTM-AE detector in a single day from the test set. Positive samples are highlighted as red circles. There are many false positive alarms during the night, which can be caused by irregular use of electricity. Once the LSTM-AE detector raises an alarm, the attack identification and MTD will be triggered. As there is no ongoing attack, the residual of the post-MTD measurement follows the χ^2 distribution. Consequently, as shown by Fig. 3.9(b), all the false positive samples have

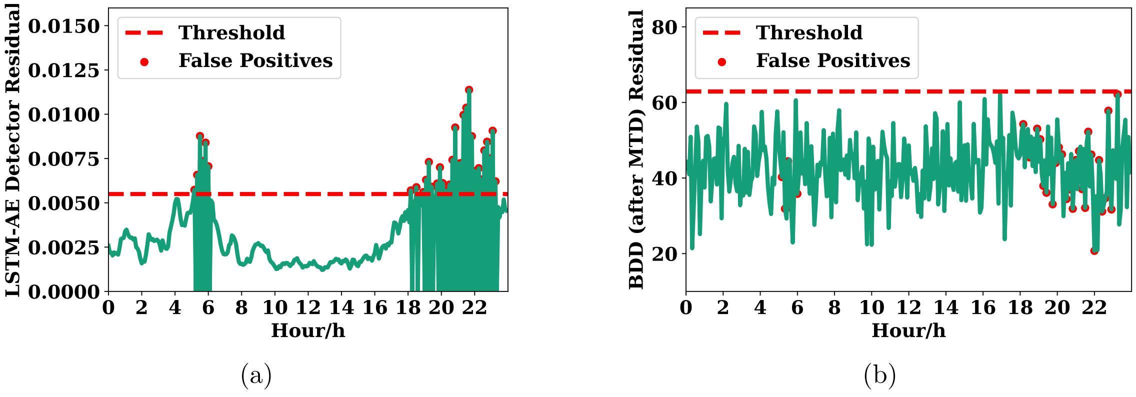


Figure 3.9: False positive rejection on LSTM-AE detector using event-triggered MTD: (a). Residual of LSTM-AE detector; and (b). Residual of BDD (possibly after MTD).

residuals lower than the BDD threshold and no further actions are needed by the system operator. On average, the FPR of the LSTM-AE detector is reduced from 12.84% to 1.84% on test set after applying DDET-MTD. Note that the MTD FPR is well controlled by the predetermined BDD FPR $\alpha = 2.0\%$.

Normal Operations

We now compare the economic performances of different MTD strategies without FDI attacks. The results in Table 3.5 demonstrate that event triggering can significantly reduce the reactance perturbation and extra operational cost of MTDs. Meanwhile, the proposed DDET-MTD has the least interference with normal system operation, making it a promising defence strategy against rare FDI attacks.

Table 3.5: Average economical performances under normal operations (in %)

Metric	**Periodic**		**Event-Triggered**		
	Max	**Robust**	**Max**	**Robust**	**DDET**
↓ Cost	0.171	0.628	0.019	0.059	0.005
↓ Reac.	27.470	45.565	3.159	5.069	2.334

3.4 Conclusion

This chapter proposes a novel Data-Driven Event-Triggered MTD (DDET-MTD) algorithm to achieve high TPR and low FPR against FDI attacks, which can benefit both the data-driven detector and the physics-based MTD. The contributions of this chapter are highlighted below.

- A novel event-triggering framework is proposed that seamlessly links the design and implementation of a data-driven detector and physics-based MTD. The proposed framework outperforms the individual approach by rejecting false positive decisions from the data-driven detector and reducing the use and cost of MTD.
- A novel measurement recovery algorithm is proposed to identify attacks through normality projection. The FDI attack detector and identifier are integrated into a single LSTM-AE deep learning model, while the physics on power flow equations is embedded to ensure the fidelity of the recovered attack.
- A bilevel optimization problem is formulated for the MTD design. In the upper level, the MTD hiddenness is improved while in the lower level, the detection accuracy is robustly guaranteed on the worst-case attack around the identified attack vector. To guarantee the feasibility and the convergence, the nonlinear non-convex bilevel optimization is further relaxed into two successive semi-definite programmings using linear matrix inequalities and duality.
- The performance of the algorithm is verified with two benchmark algorithms under periodic and event trigger settings, using real-time load and solar profiles.

Part II

Integrated Learning and Optimization for Load Forecasting and Operations

Chapter 4

E2E-AT: A Unified Framework for Tackling Uncertainty in Task-aware End-to-end Learning

4.1 Introduction

Successful machine learning involves a complete pipeline of data, model, and downstream applications. Instead of treating them separately, there has been a prominent increase of attention within the constrained optimization (CO) and machine learning (ML) communities towards combining prediction and optimization models. The so-called end-to-end (E2E) learning captures the task-based objective for which they will be used for decision making. The idea of integrating learning and model has been applied to many critical industrial activities, such as better management of the power system [119, 123], constraint satisfaction in control [149], and routing behavior of the transportation network [150]. Therefore, it is essential to understand the vulnerability of this learning framework and to study the associated robust enhancement.

The parameters in CO and task-aware cost can be classified into predictable and unpredictable parameters. Here, the predictable parameters are those that can be modeled by the forecast distribution, while the unpredictable parameters cannot. In power system operation, the predictable parameters are the load and renewable energy while the unpredictable parameters are the system configurations, such as generator cost, thermal constraints, and susceptance of transmission lines. Conventionally in ML, the uncertainty of the data is only defined by the input of the forecast model, the worst case of which can be found by adversarial attack and treated by adversarial training [151]. In the E2E model, the definition of data is augmented by the CO parameter[1]. Although the uncertainty of predictable parameters can be modeled by the forecast model, the uncertainty of unpredictable parameters is usually overlooked in the literature. The authors in [124] also realize that dynamic environments, such as varying parameters, cannot be adapted to current E2E learning. However, they treat the varying parameters as input to the forecast model, which may not be realistic since the unpredictable parameters may not be realized when forecasting.

In this chapter, a multi-level optimization problem is constructed to systematically study the E2E learning where the down-stream decision-makings are modeled as lower-level problems. It is demonstrated that a single-stage formulation cannot reflect real-time operation and lead to a deviation from the optimal decision. In the meantime, neglecting the uncertainty of unpredictable parameters during training causes a new trigger for generalization errors. The main contribution of this chapter is to treat multiple sources of uncertainty from both input space of ML and unpredictible parameter in COs uniformly as a robust optimization problem, which can be practically solved by end-to-end adversarial training (E2E-AT) and differentiable layers. When forecaster is piece-wise linear and the down-stream optimizations are quadratic, an exact input space adversarial attack is proposed to increase the task-aware cost as an augmented integer programming. Our method can be viewed as a natural interconnection between adversarial training, the implicit layer, and E2E learning. Finally, the performance of E2E-AT is evaluated using a real-world end-to-end power system operation problem, including load forecasting and sequential scheduling tasks.

[1] The unpredictable parameter of task-aware cost is merged as part of CO.

4.2 Preliminaries

4.2.1 Adversarial Training and Certified Robustness

In supervised learning, a parametric model $\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta})$ can be defined to map from input $\boldsymbol{x}$ to label $\boldsymbol{y}$ for $(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{D}$ by minimizing the following empirical loss:

$$\min_{\boldsymbol{\theta}} \sum_{i\in\mathcal{D}} \mathcal{L}(\boldsymbol{f}(\boldsymbol{x}^i;\boldsymbol{\theta}),\boldsymbol{y}^i) \tag{4.1}$$

In this chapter, we use $(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{D}$ and $i \in \mathcal{D}$ interchangeably to denote a sample of the dataset.

It has been well studied that the deep neural network is prone to small perturbations on its input. In security-critical applications, robustness has become an emerging factor. Adversarial training has been shown to improve the robustness of NN, which transforms (4.1) into a robust optimization [151]:

$$\min_{\boldsymbol{\theta}} \sum_{i\in\mathcal{D}} \max_{\boldsymbol{\delta}_i\in\Delta} \mathcal{L}(\boldsymbol{f}(\boldsymbol{x}^i+\boldsymbol{\delta}^i;\boldsymbol{\theta}),\boldsymbol{y}^i) \tag{4.2}$$

where $\Delta = \{\boldsymbol{\delta} : \|\boldsymbol{\delta}\|_\infty \leq \epsilon\}$ is the attack budget set for some small $\epsilon > 0$.

Robust optimization (4.2) is practically solved by iterative approaches in which adversarial perturbation is first solved by inner minimization through gradient ascent. To keep the attack within Δ, projected gradient descent (PGD) [151] is adopted:

$$\boldsymbol{\delta}^i_{t+1} = \mathcal{P}_\Delta\left(\boldsymbol{\delta}^i_t + \gamma \cdot \operatorname{sign}\left(\nabla_{\boldsymbol{\delta}}\mathcal{L}\left(\boldsymbol{f}\left(\boldsymbol{x}^i+\boldsymbol{\delta}^i_t\right),\boldsymbol{y}^i\right)\right)\right) \tag{4.3}$$

where $\mathcal{P}_\Delta$ is the projector on Δ and γ is the step size.

Although adversarial training is effective in improving robustness, it may give the wrong sign of security as inner maximization is inexactly solved. A certification approach can be made on the feedforward neural network with ReLU activations, e.g., a piecewise linear neural network[2]. Using the big-M method, the neural network with d layers can be

[2] Convolution layer and other piecewise linear activations, such as leaky ReLU, are also piecewise linear.

represented by the following set [152]

$$\mathcal{C}_{\text{nn}}(\boldsymbol{x};\boldsymbol{\theta}) = \left\{ \boldsymbol{y} : \begin{array}{l} \boldsymbol{z}_1 = \boldsymbol{x}, \boldsymbol{z}_{i+1} \geq \boldsymbol{W}_i \boldsymbol{z}_i + \boldsymbol{b}_i, \\ \boldsymbol{z}_{i+1} \geq \boldsymbol{0}, \boldsymbol{u}_i \cdot \boldsymbol{v}_i \geq \boldsymbol{z}_{i+1} \\ \boldsymbol{W}_i \boldsymbol{z}_i + \boldsymbol{b}_i \geq \boldsymbol{z}_{i+1} + (\boldsymbol{1} - \boldsymbol{v}_i)\boldsymbol{l}_i, \\ \boldsymbol{v}_i \in \{0,1\}^{|\boldsymbol{v}_i|}, \quad i = 1, \cdots, d-2 \\ \boldsymbol{y} = \boldsymbol{W}_{d-1}\boldsymbol{z}_{d-1} + \boldsymbol{b}_{d-1} \end{array} \right\} \tag{4.4}$$

where $\boldsymbol{\theta}_i = (\boldsymbol{W}_i, \boldsymbol{b}_i)$ is the weight and bias of the i-th layer. $\boldsymbol{u}_i$ and $\boldsymbol{l}_i$ are the upper and lower bounds of the output of i-th layer, which can be efficiently estimated by interval bound propagation (IBP) [153]. $\boldsymbol{v}_i$ is an integer vector that controls the activation of ReLUs.

Based on (4.4), the inner maximization of (4.2) becomes:

$$\begin{aligned} \max\nolimits_{\boldsymbol{\delta}^i} \quad & \mathcal{L}(\hat{\boldsymbol{y}}^i, \boldsymbol{y}^i) \\ \text{subject to} \quad & \boldsymbol{\delta}^i \in \Delta, \hat{\boldsymbol{y}}^i \in \mathcal{C}_{\text{nn}}(\boldsymbol{x}^i + \boldsymbol{\delta}^i; \boldsymbol{\theta}) \end{aligned} \tag{4.5}$$

As $\mathcal{C}_{\text{nn}}(\boldsymbol{x};\boldsymbol{\theta})$ is defined by mixed integer linear constraints, (4.5) can be solved exactly for certain types of loss function. When cross-entropy loss is used, we can choose to maximize the output of each logit, and if all the logits are not greater than the ground-truth logit within the attack budget, the NN is said to be certified robust at the candidate sample. For a regression task, (4.5) can be formulated as mixed-integer linear programming (MILP) to maximize or minimize the forecast value [154].

4.2.2 End-to-End Machine Learning

End-to-end learning takes the fusion of prediction (ML) and decision making (CO), which aims to find the map from the data to the optimal decision such that the learned ML model can optimally reflect the CO through training. Meanwhile, E2E learning distinguishes itself from separate supervised learning followed by the CO in which the prediction divergence and the task-aware cost can have a large mismatch [16].

Referring to a recent review [155], the CO can be modeled as

$$\begin{aligned} \boldsymbol{z}^\star = \operatorname{argmin}_{\boldsymbol{z}} \quad & \ell(\boldsymbol{z};\boldsymbol{y}) \\ \text{subject to} \quad & \boldsymbol{z} \in \mathcal{C}(\boldsymbol{z};\boldsymbol{y}) \end{aligned} \tag{4.6}$$

where $\ell(\cdot)$ is the objective function, $\boldsymbol{z}$ is the decision variable, $\boldsymbol{y}$ is the parameter, and $\mathcal{C}$ is the constraint set.

Let $\hat{\boldsymbol{y}}$ be the output of $\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta})$ and let $\hat{\boldsymbol{z}}^\star$ be the minimizer of (4.6) parameterized by $\hat{\boldsymbol{y}}$. There are two options to learn the E2E model guided by (4.6). In the supervised learning setting, a suitable loss function $\mathcal{M}$ can be chosen to minimize the difference to ground-truth decision, e.g. $\mathcal{M}(\hat{\boldsymbol{z}}^\star, \boldsymbol{z}^\star)$ or $\mathcal{M}(\ell(\hat{\boldsymbol{z}}^\star;\hat{\boldsymbol{y}}), \ell(\boldsymbol{z}^\star;\boldsymbol{y}))$ [156] . Alternatively, the regret function [21] can be implemented:

$$\text{regret} = \ell(\hat{\boldsymbol{z}}^\star;\hat{\boldsymbol{y}}) - \ell(\boldsymbol{z}^\star;\boldsymbol{y}) \tag{4.7}$$

The ground truth decision $\boldsymbol{z}^\star$ in (4.7) is not necessary to compute exactly, since $\ell(\boldsymbol{z}^\star;\boldsymbol{y})$ is constant [22]. Therefore, we argue that the regret function (4.7) is more intrinsic, as the training loss has the same format as the CO objective (4.6).

4.3 A Heuristic Example

Before we move into a detailed formulation of the robustness of E2E learning, we first highlight a misleading formulation, which violates the intention of E2E learning and introduces model uncertainties between training and inference.

Consider two E2E learning problems based on the supervised setting:

$$\begin{aligned} \operatorname{argmin}_{\boldsymbol{\vartheta},\hat{\boldsymbol{z}}} \quad & M(\hat{\boldsymbol{z}}) := \textstyle\sum_{i\in\mathcal{D}} \mathcal{M}(\ell(\hat{\boldsymbol{z}}^i;\hat{\boldsymbol{y}}^i), \ell(\boldsymbol{z}^{i\star};\boldsymbol{y}^i)) \\ \text{subject to} \quad & \hat{\boldsymbol{z}}^i \in \mathcal{C}(\boldsymbol{z}^i;\hat{\boldsymbol{y}}^i), \quad i \in \mathcal{D} \end{aligned} \tag{4.8}$$

whose minimum point is denoted as $(\boldsymbol{\theta}_1^\star, \hat{\boldsymbol{z}}_1^\star)$. And,

$$\begin{aligned} \text{argmin}_{\boldsymbol{\vartheta},\hat{\boldsymbol{z}}} \quad & M(\hat{\boldsymbol{z}}) := \textstyle\sum_{i \in \mathcal{D}} \mathcal{M}(\ell(\hat{\boldsymbol{z}}^i; \hat{\boldsymbol{y}}^i), \ell(\boldsymbol{z}^{i\star}; \boldsymbol{y}^i)) \\ \text{subject to} \quad & \hat{\boldsymbol{z}}^i \in \text{argmin}_{\boldsymbol{z}^i} \{\ell(\boldsymbol{z}^i; \hat{\boldsymbol{y}}^i) : \boldsymbol{z}^i \in \mathcal{C}(\boldsymbol{z}^i; \hat{\boldsymbol{y}}^i)\}, \quad i \in \mathcal{D} \end{aligned} \tag{4.9}$$

with minimum point $(\boldsymbol{\theta}_2^\star, \hat{\boldsymbol{z}}_2^\star)$.

E2E learning problem (4.8) takes the sample-wise constraints of (4.6) as its constraints while (4.9) is a bilevel optimization problem whose upper level minimizes the difference to the ground-truth objective $\ell(\boldsymbol{z}^{i\star}; \boldsymbol{y}^i)$ and each lower level solves the sample-wise decision-making problem in parallel.

In the **inference stage**, we first predict $\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \boldsymbol{\theta}^\star)$ and then solve the optimization problem (4.6) parameterized by $\hat{\boldsymbol{y}}$. The optimal decision variables are denoted as $\hat{\boldsymbol{z}}_r^\star(\boldsymbol{\theta}_1^\star)$ and $\hat{\boldsymbol{z}}_r^\star(\boldsymbol{\theta}_2^\star)$, respectively.

Proposition 7. *Given two formulations* (4.8) *and* (4.9) *of E2E learning,* $M(\hat{\boldsymbol{z}}_1^\star) \leq M(\hat{\boldsymbol{z}}_2^\star) = M(\hat{\boldsymbol{z}}_r^\star(\boldsymbol{\theta}_2^\star)) \leq M(\hat{\boldsymbol{z}}_r^\star(\boldsymbol{\theta}_1^\star))$.

The proof can be found in Appendix B.8.

It shows that formulation (4.8) can result in different decisions at the training and inference stages. Therefore, Proposition 7 demonstrates that a misformulation of CO can possibly lead to misled decision making at inference. Broadly speaking, the ignorance of the optimization model contributes to a new source of uncertainties, causing generalization error, in addition to the well-studied uncertainties in the dataset (e.g. out-of-distribution samples, adversarial attack, etc.). This motivates us to take into account the uncertainties in both the input space and the CO formulation.

4.4 Sequential E2E Learning as Multi-Level Optimization

Proposition 7 implies formulating E2E learning as a multi-level optimization problem by respecting the sequence of downstream tasks. Given $\boldsymbol{\theta}$, sequential decision makings can be denoted as lower-level problems after the prediction.

Formally, at the inference stage, the cost of one-time decision making on $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$ can be written as:

$$
\begin{aligned}
\min \quad & \mathcal{L}(\hat{\boldsymbol{z}}_1, \cdots, \hat{\boldsymbol{z}}_m; \boldsymbol{y}, \boldsymbol{\phi}_0) \\
\text{subject to} \quad & \hat{\boldsymbol{z}}_i \in \operatorname{argmin}_{\boldsymbol{z}_i} \{\ell_i(\boldsymbol{z}_i; \boldsymbol{y}, \boldsymbol{\phi}_i) : \boldsymbol{z}_i \in \mathcal{C}_i(\boldsymbol{z}_i; \boldsymbol{y}, \hat{\boldsymbol{z}}_{i-1}, \boldsymbol{\phi}_i)\}, \quad i = 2, \cdots, m \\
& \hat{\boldsymbol{z}}_1 \in \operatorname{argmin}_{\boldsymbol{z}_1} \{\ell_1(\boldsymbol{z}_1; \boldsymbol{y}, \boldsymbol{\phi}_1) : \boldsymbol{z}_1 \in \mathcal{C}_1(\boldsymbol{z}_1; \boldsymbol{y}, \hat{\boldsymbol{y}}, \boldsymbol{\phi}_1)\} \\
& \hat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\theta})
\end{aligned}
\tag{4.10}
$$

Inference (4.10) contains m downstream tasks with objective $\ell_i(\cdot)$s, constraint set $\mathcal{C}_i(\cdot)$s, and **unpredictable** parameter $\boldsymbol{\phi}_i$s. It can be compactly denoted as

$$
\begin{aligned}
\min_{\hat{\boldsymbol{z}}} \quad & \mathcal{L}(\hat{\boldsymbol{z}}; \boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\phi}) \\
\text{subject to} \quad & \hat{\boldsymbol{z}} \in \mathcal{C}_{\text{E2E}}(\boldsymbol{z}; \boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\phi})
\end{aligned}
\tag{4.11}
$$

where $\hat{\boldsymbol{z}} = [\hat{\boldsymbol{z}}_1, \cdots, \hat{\boldsymbol{z}}_m]$ and $\boldsymbol{\phi} = [\boldsymbol{\phi}_0, \cdots, \boldsymbol{\phi}_m]$. $\mathcal{C}_{\text{E2E}}(\cdot)$ represents the constraint set representing all constraints of (4.10). Note that we consider the COs as parametric functions that can be uniquely modeled by $(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\phi})$.

To train $\boldsymbol{f}(\cdot; \boldsymbol{\theta})$, the empirical training loss can be minimized:

$$
\begin{aligned}
\min_{\boldsymbol{\theta}} \quad & \textstyle\sum_{i \in \mathcal{D}} \mathcal{L}(\hat{\boldsymbol{z}}^i; \boldsymbol{\theta}, \boldsymbol{x}^i, \boldsymbol{y}^i, \boldsymbol{\phi}) \\
\text{subject to} \quad & \hat{\boldsymbol{z}}^i \in \mathcal{C}_{\text{E2E}}(\boldsymbol{z}^i; \boldsymbol{\theta}, \boldsymbol{x}^i, \boldsymbol{y}^i, \boldsymbol{\phi}), \quad i \in \mathcal{D}
\end{aligned}
\tag{4.12}
$$

Since this chapter does not focus on solving optimizations, we restrict down-stream optimizations to quadratic programming (QP). QP has been widely implemented in many industrial applications and is mostly discussed in the E2E learning literature [155]. Meanwhile, since QP is convex and if the Slater condition holds, the Karush–Kuhn–Tucker (KKT) condition is sufficient for optimality [157]. Therefore, optimizations at the lower level can be replaced by the corresponding KKT conditions, known as the mathematical program with equilibrium constraints (MPEC) [158]. Therefore, if a linear parametric model is considered, it is possible to solve (4.12) exactly using optimization software.

In addition to the linear model, stochastic gradient descent (SGD) needs to be applied on the mini-batches of $\mathcal{D}$ to train the NN model. SGD requires 1) a forward pass in which the optimizations are solved and 2) a backward pass to update the NN parameters. Denote the equality part of KKT condition of the i-th optimization as

$$\boldsymbol{g}(\boldsymbol{z}_i^\star; \hat{\boldsymbol{z}}_{i-1}, \boldsymbol{y}, \boldsymbol{\phi}_i) = \mathbf{0} \tag{4.13}$$

which can be viewed as differentiable layer (OptNet) [105] by the implicit function theorem [104]:

$$\frac{\partial \boldsymbol{z}_i^\star}{\partial \hat{\boldsymbol{z}}_{i-1}} = -\left(\frac{\partial \boldsymbol{g}(\boldsymbol{z}_i^\star; \hat{\boldsymbol{z}}_{i-1}, \boldsymbol{y}, \boldsymbol{\phi}_i)}{\partial \boldsymbol{z}_i}\right)^{-1} \frac{\partial \boldsymbol{g}(\boldsymbol{z}_i^\star; \hat{\boldsymbol{z}}_{i-1}, \boldsymbol{y}, \boldsymbol{\phi}_i)}{\partial \hat{\boldsymbol{z}}_{i-1}} \tag{4.14}$$

Note that in (4.13) and (4.14), the optimal dual variables $(\boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$ in the KKT conditions are omitted for simplicity. As long as the Jacobian matrix is not singular, the gradient of the output $\boldsymbol{z}_i^\star$ to the input $\hat{\boldsymbol{z}}_{i-1}$ exists, allowing backpropagation through the differentiable layers.

4.5 Unified Robustness Framework

When treating the E2E framework as an integrated model, the data source includes both conventionally defined data samples $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$ and the unpredictable parameter $\boldsymbol{\phi}$ of COs. Small input uncertainties have been shown to cause a significant performance drop, and it is reasonable to draw a similar conclusion for the unpredictable parameter. In fact, Proposition 7 shows that any mismatches between the COs used for training and inference should be explicitly considered. Although COs can take an infinite number of formulations, without loss of generality, we restrict the uncertainties of COs in the parameters of objective and constraints. We argue that the unpredictable parameter used during training may not be the same as that used for real-time decision-making. For example, in power system operation, the production costs of the generators can vary over time, and the resistance and susceptance of transmission lines can be altered both intentionally and unintentionally. These parameters are usually not known to the system operator in advance or at least not fully aware when training the forecast model. See Fig.

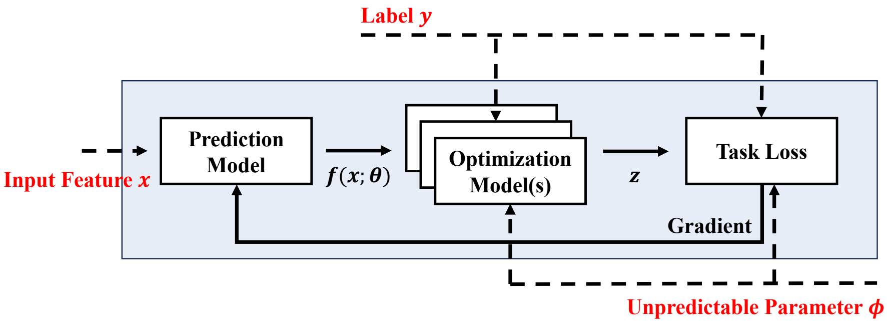


Figure 4.1: An illustration of E2E-AT learning. We consider the uncertainties in the input data $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$ and the uncertainties in the COs, specifically, the unpredictable parameter $\boldsymbol{\phi}$.

4.1 for an illustration.

4.5.1 Formulation

Consider the uncertainty of the input $\boldsymbol{x}+\boldsymbol{\delta}_x \in \mathcal{X}$ and the unpredictable parameter $\boldsymbol{\phi}+\boldsymbol{\delta}_\phi \in \Phi$. Denote $\boldsymbol{\psi} = (\boldsymbol{x}, \boldsymbol{\phi}) \in \Psi := \mathcal{X} \times \Phi$. The worst scenario, which maximizes the task-aware objective, can be formulated as

$$\begin{aligned} \max_{\boldsymbol{\psi}\in\Psi} \quad & \mathcal{L}(\hat{\boldsymbol{z}}; \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{y}) \\ \text{subject to} \quad & \hat{\boldsymbol{z}} \in \mathcal{C}_{\text{E2E}}(\boldsymbol{z}; \boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{y}) \end{aligned} \tag{4.15}$$

in which $\boldsymbol{\theta}$ is fixed.

Consequently, a robust optimization can be formulated. A unified E2E adversarial training (E2E-AT) considering both input and CO uncertainties becomes

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \textstyle\sum_{i\in\mathcal{D}} \max_{\boldsymbol{\psi}^i\in\Psi^i} \mathcal{L}(\{\hat{\boldsymbol{z}}^i; \boldsymbol{\theta}, \boldsymbol{\psi}^i, \boldsymbol{y}^i) \\ \text{subject to} \quad & \hat{\boldsymbol{z}}^i \in \mathcal{C}_{\text{E2E}}(\boldsymbol{z}^i; \boldsymbol{\theta}, \boldsymbol{\psi}^i, \boldsymbol{y}^i), \quad i \in \mathcal{D} \end{aligned} \tag{4.16}$$

where the constrains are subject to both minimization and maximization. Adversarial training can be adopted to solve (4.16). Similarly to the implicit function theorem (4.14), the gradient of the constraint exists, regardless of the minimization or maximization. Therefore, Danskin's theorem can be used by first solving the inner maximization through gradient ascent (with $\boldsymbol{\theta}$ fixed) and then for the outer gradient descent (with $\boldsymbol{\psi}$ fixed). Although using the Danskin theorem requires one to exactly solve the inner maximization, it can give a descent direction for suboptimal $\boldsymbol{\psi}$, e.g. solved by PGD (4.3) and is applicable to various adversarial training algorithms [151, 159, 160].

4.5.2 Certified Robustness

Although PGD (4.3) is effective for E2E-AT, it cannot verify the robustness, as it only finds the local maximum [161]. In particular, a robustness certification on (4.11) verifies if an adversarial example exists within the budget Δ such that the task-aware objective is altered by a certain amount. The key is to find the exact adversarial attack in (4.15). We show that for specific type of objective and COs (e.g. affine-parametric QPs), optimal solution to (4.15) can be solved exactly, which extends the certified robustness in piecewise linear neural network (4.4).

Proposition 8. *The affine-parametric QP:*

$$\begin{aligned} \boldsymbol{z}_{i+1} := \quad & \arg\min_{\boldsymbol{z}} \quad \tfrac{1}{2}\boldsymbol{z}^T\boldsymbol{Q}\boldsymbol{z} + \boldsymbol{q}^T\boldsymbol{z} \\ & \text{subject to} \quad \boldsymbol{A}\boldsymbol{z} + \boldsymbol{G}\boldsymbol{z}_i \leq \boldsymbol{b} \\ & \qquad\qquad\quad \boldsymbol{C}\boldsymbol{z} + \boldsymbol{H}\boldsymbol{z}_i = \boldsymbol{d} \end{aligned} \tag{4.17}$$

can be equivalently written as the set of mixed integer linear constraints:

$$\boldsymbol{Q}\boldsymbol{z}_{i+1} + \boldsymbol{q} + \boldsymbol{A}^T\boldsymbol{\lambda}_{i+1} + \boldsymbol{C}^T\boldsymbol{\nu}_{i+1} = 0$$

$$\boldsymbol{C}\boldsymbol{z}_{i+1} + \boldsymbol{H}\boldsymbol{z}_i - \boldsymbol{d} = \boldsymbol{0}$$

$$\boldsymbol{0} \leq \boldsymbol{\lambda}_{i+1} \leq \boldsymbol{\varphi}\boldsymbol{M}$$

$$(\boldsymbol{\varphi} - \boldsymbol{1})\boldsymbol{M} \leq \boldsymbol{A}\boldsymbol{z}_{i+1} + \boldsymbol{G}\boldsymbol{z}_i - \boldsymbol{b} \leq \boldsymbol{0}$$

$$\boldsymbol{\varphi} \in \{0, 1\}^{|\boldsymbol{\varphi}|}$$

where $\boldsymbol{Q}, \boldsymbol{q}, \boldsymbol{A}, \boldsymbol{G}, \boldsymbol{b}, \boldsymbol{C}, \boldsymbol{H}, \boldsymbol{d}$ *are the parameters with proper dimensions.* $\boldsymbol{\varphi}$ *is binary vector and* $\boldsymbol{M}$ *is a large positive number. The equalities and inequalities are element-wise.*

The proof can be found in the Appendix B.9.

Due to the complexity of integer programming, we restrict the original settings in [105] by assuming linearity in uncertain terms for certified robustness. For example, when the uncertainty in the input feature is considered, the parameter $\boldsymbol{z}_i$ that represents the optimal decision of the previous task is decoupled from the variable $\boldsymbol{z}$ and is affine so that the above reformulation is linear. When the uncertainty of CO is considered, we assume that the uncertain unpredictable parameter is decoupled from the decision variables as well. We note that this setting follows the disciplined parametrized programming (DPP) [106].

Since both affine-parametric QP (4.17) and the neural network (4.4) can be represented by mixed-integer linear constraints, maximizing an affine function subject to these constraints becomes mixed-integer linear programming (MILP), which can be effectively solved and used to certify the worst possible cost.

4.5.3 Discussion on the Robustness

Previously, the uncertainty involved in COs has been considered in many E2E learning algorithms. From the perspective of contextual optimization, E2E learning applies ML to predict the uncertain parameter [16]. In such setting, probabilistic forecast and stochastic program can be implemented [19]. However, we view the entire E2E learning as an integrated model such that the uncertainty of the intermediate variable can be merged within the E2E training objective. Alternatively, we separate the parameters of COs into two parts. The first part (predictable parameter) is forecasted by the ML while the second part is unpredictable (unpredictable parameter). Indirectly, the uncertainty of the predictable parameter is tackled by adversarial training on the input feature, while the uncertainty of unpredictable parameter needs to be treated as well.

We note that the physical meaning of the uncertainty of the unpredictable parameter is different from that of the adversarial attack in the input feature space. Although the ultimate goal is to improve the robustness of the ML, the adversarial attack assumes that there exists a malicious party that can find the worst attack. However, the uncertainty of the unpredictable parameter always exists regardless of the malicious party. Inspired by the previous work [4, 162], we can alternatively view the E2E-AT on the unpredictable parameter as the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \alpha \textstyle\sum_{i\in\mathcal{D}} \mathcal{L}(\bar{\boldsymbol{z}}^i; \boldsymbol{\theta}, \boldsymbol{x}^i, \boldsymbol{y}^i, \bar{\boldsymbol{\phi}}) + (1-\alpha) \textstyle\sum_{i\in\mathcal{D}} \mathbf{E}_{(\hat{\boldsymbol{z}}^i, \boldsymbol{\phi}^i)}[\mathcal{L}(\hat{\boldsymbol{z}}^i; \boldsymbol{\theta}, \boldsymbol{x}^i, \boldsymbol{y}^i, \boldsymbol{\phi}^i)] \\ \text{subject to} \quad & \bar{\boldsymbol{z}}^i \in \mathcal{C}_{\text{E2E}}(\boldsymbol{z}^i; \boldsymbol{\theta}, \boldsymbol{x}^i, \boldsymbol{y}^i, \bar{\boldsymbol{\phi}}), \quad i \in \mathcal{D} \\ & \hat{\boldsymbol{z}}^i \in \mathcal{C}_{\text{E2E}}(\boldsymbol{z}^i; \boldsymbol{\theta}, \boldsymbol{x}^i, \boldsymbol{y}^i, \boldsymbol{\phi}^i), \quad \boldsymbol{\phi}^i \in \Phi^i, i \in \mathcal{D} \end{aligned} \tag{4.19}$$

where $\bar{\boldsymbol{z}}^i$ is the decision variable from the COs parameterized by the nominal unpredictable parameter $\bar{\boldsymbol{\phi}}$.

It can be argued that the NN forecaster is trained by considering the expected task-aware cost due to uncertainties. The learning objective (4.19) takes the nominal unpredictable parameter (denoted as $\bar{\boldsymbol{\phi}}$) and the expected uncertainties over $\boldsymbol{\Phi}^i$ into account, which are balanced by the hyperparameter α. As shown in (4.19), this stochastic program can be solved by sampling $\boldsymbol{\phi}^i \in \boldsymbol{\Phi}^i$ during training. In addition, if the uncertainty set of the unpredictable parameter is independent of the sample, $\boldsymbol{\Phi}$ is not subject to index i.

4.5.4 Final Training Objective

In E2E-AT, (4.19) is solved by robust optimization by finding the maximum over $\boldsymbol{\psi}^i \in \Psi^i$, as in (4.16). We also take the input uncertainty into account:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \alpha \cdot \textstyle\sum_{i\in\mathcal{D}} \mathcal{L}(\bar{\boldsymbol{z}}^i; \boldsymbol{\theta}, \bar{\boldsymbol{\psi}}^i, \boldsymbol{y}^i) + (1-\alpha) \cdot \textstyle\sum_{i\in\mathcal{D}} \max_{\boldsymbol{\psi}^i \in \Psi^i} \mathcal{L}(\hat{\boldsymbol{z}}^i; \boldsymbol{\theta}, \boldsymbol{\psi}^i, \boldsymbol{y}^i) \\ \text{subject to} \quad & \bar{\boldsymbol{z}}^i \in \mathcal{C}_{\text{E2E}}(\boldsymbol{z}^i; \boldsymbol{\theta}, \bar{\boldsymbol{\psi}}, \boldsymbol{y}^i) \\ & \hat{\boldsymbol{z}}^i \in \mathcal{C}_{\text{E2E}}(\boldsymbol{z}^i; \boldsymbol{\theta}, \boldsymbol{\psi}^i, \boldsymbol{y}^i), \quad i \in \mathcal{D} \end{aligned} \tag{4.20}$$

The new adversarial training objective provides an upper bound on the expectation part of (4.19). Meanwhile, similar to adversarial training on image tasks [159], α can be used to balance the clean and adversarial accuracies. When $\alpha \to 1$, (4.20) becomes the original E2E learning and when $\alpha \to 0$, it becomes pure adversarial training. In addition, clean and adversarial training losses may not directly reflect clean and robust accuracy for image tasks, causing an unbalanced training objective. In E2E-AT, the two objectives are defined by the task, which is the exact metric during decision making.

Previously in [4, 162], the authors reformulate a bilevel optimization problem into robust optimization and use the implicit function theorem for constraint satisfaction. Connecting (4.19) to (4.20), we extend [4] to training ML models. It can be seen that for each mini-batch in E2E-AT, a similar robust optimization is solved as [4]. The implicit function theorem is also used to learn the task-aware objective while satisfying the constraints.

4.5.5 'Free' E2E Adversarial Training

In E2E-AT, the number of forward and backward passes is equal to $\texttt{no\_batch} \times (\texttt{no\_pgd} + 1) \times \texttt{no\_epoch}$ (assuming that all minibatches have the same size). Adversarial training is computationally ineffective due to intensive backpropagation (controlled by the complexity of the neural network). The computational burden is even higher in E2E-AT as in each forward pass, the COs need to be solved (controlled by the complexity of the COs). To save training time, we adopt the gradient reuse strategy. In the *adversarial training for free* [163], the attack vector and the model parameter are repeatedly updated in the same mini-batch for `no_pgd` times. Then, `epoch_no` is divided by `no_pgd` to maintain the total number of model updates unchanged. This results in $\texttt{no\_batch} \times \texttt{no\_epoch}$ numbers of forward and backward passes, which is the same as the clean E2E training.

4.6 Experiment

In the experiment, the NN is trained to forecast the load in the power system. The robustness of various E2E-AT settings is explored. Detailed experiment settings can be found in Appendix E.3 and more results can be found in the Appendix E.4.

4.6.1 Power System Operations

A practical power system operation problem, named as network constrained economic dispatch (NCED), is considered, which has been widely used in the US and can be formulated as two-stage QP or LP [25]. In stage one (also known as dispatch), the set points of the generator are determined based on the **forecast** load. The goal of the first stage is to minimize the generator cost while meeting the physical constraints of the grid. When the generator has been dispatched, we consider a realization on the actual load by solving the second stage problem (also known as redispatch), in which any mismatches on the load and generation from stage one, as well as the violation of the physical constraints of the grid, will be penalized by extra cost:

$$\boldsymbol{P}_g^\star = \text{Dispatch}(\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}), \boldsymbol{b}) \tag{4.21a}$$

$$\boldsymbol{P}_{ls}^\star, \boldsymbol{P}_{gs}^\star = \text{Redispatch}(\boldsymbol{y}, \boldsymbol{P}_g^\star, \boldsymbol{b}) \tag{4.21b}$$

where $\boldsymbol{P}_g^\star$ is the optimal generator dispatch, $\boldsymbol{P}_{ls}^\star$ is the load shedding and $\boldsymbol{P}_{gs}^\star$ is the power storage. $\boldsymbol{b}$ is the susceptance of the transmission line (when the resistance is close to zero, the susceptance is reciprocal to the reactance). The task-aware objective is defined as

$$\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{c}_g^T \boldsymbol{P}_g + \boldsymbol{c}_{ls}^T \boldsymbol{P}_{ls} + \boldsymbol{c}_{gs}^T \boldsymbol{P}_{gs} \tag{4.22}$$

where $\boldsymbol{c}_g$, $\boldsymbol{c}_{ls}$, and $\boldsymbol{c}_{gs}$ are the coefficients such that $\boldsymbol{c}_{ls} \gg \boldsymbol{c}_{gs} \gg \boldsymbol{c}_g$. That is, the load shedding is more costly. The detailed formulation on the dispatch and re-dispatch problem used for this section can be found in Appendix E.1.

4.6.2 Training Settings

We use an open source load forecasting dataset from the Texas Backbone Power System [164] on a modified IEEE bus-14 system. We randomly collect 1.0k samples and use a feedforward neural network with three hidden layers to forecast the load[3]. We do E2E-AT with **a).** input feature uncertainties $\boldsymbol{\delta}_x$, **b).** uncertainty of the unpredictable parameters $\boldsymbol{\delta}_\phi$, and **c).** integrated uncertainties of both $(\boldsymbol{\delta}_x, \boldsymbol{\delta}_\phi)$. We first implement natural (or clean) E2E learning, based on which we warm-start the E2E-ATs. Adam optimizer is used, and 'Adversarial training for free' [163] is applied to reuse the gradients for PGD.

In detail, **a).** Since meteorological features have been normalized into $[0, 1]$, we attack with a normalized budget $\epsilon_x \in \{0.02, 0.03\}$. The inner maximization is solved with 7 PGD steps and the step size is dynamically set as $\epsilon_x/7 \times 2$. We summarize this setting from [159]. We clamp the attacked input into $[0, 1]$ whenever it is updated. **b).** We consider uncertainties on the susceptance $\boldsymbol{b}$ in the **redispatch** problem. Since each transmission line can have different nominal susceptance, we set the budget $\epsilon_\phi \in \{0.05, 0.15\}$ as the proportion to the individual nominal value, which is consistent with the common operation range of susceptance [74]. **c).** We do E2E-AT with $(\epsilon_x, \epsilon_\phi) \in \{(0.02, 0.05), (0.03, 0.15)\}$ for the integrated uncertainties. The other settings are the same as in (a) and (b).

4.6.3 Performance of E2E-AT

Multi-run adversarial attacks are evaluated in Table 4.1. For each sample and attack scenario, we randomly select three starting points within the attack budget and report the **worst** task-aware cost (4.22) to reduce the variance. Specifically, five training algorithms are compared:

- **NAT**: Natural training with task-aware loss;
- **AT-MSE**: Adversarial training with MSE loss;
- **AT-INPUT**: Adversarial training with task-aware loss and input uncertainties;
- **AT-PARA**: Adversarial training with task-aware loss and unpredictable CO uncertainties; and
- **AT-BOTH**: Adversarial training with task-aware loss and integrated uncertainties.

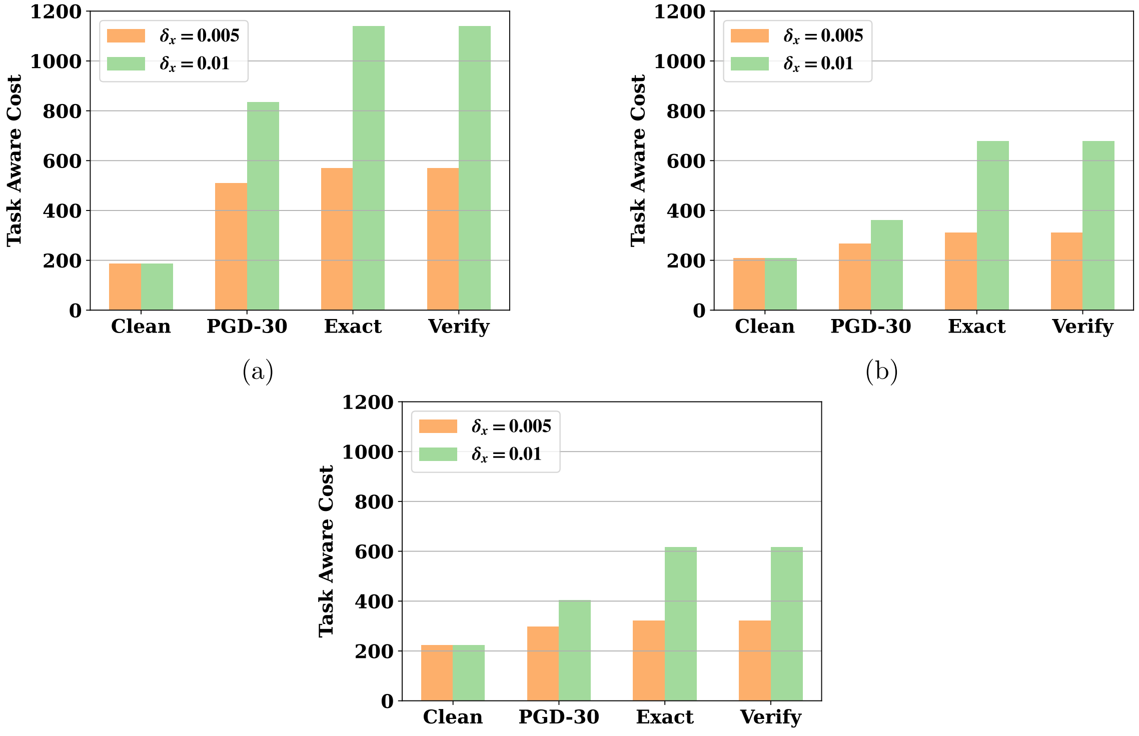
[3] More experiment can be found in the Appendix E.4

Table 4.1: Performances of the E2E-AT.

Training Method			**Clean**	**Input Attack**, ϵ_x			**CO Attack**, ϵ_ϕ			**Integrated Attack**, $(\epsilon_x, \epsilon_\phi)$		
ϵ_x	ϵ_ϕ	α	N/A	0.01	0.02	0.03	0.05	0.10	0.15	(0.01,0.05)	(0.02,0.10)	(0.03,0.15)
NAT: Natural Training with Task Loss												
0	0	0	201.6	653.6	1188.5	1792.1	754.6	2044.3	3468.4	1153.9	3090.8	5214.4
AT-MSE: Adversarial Training with MSE Loss												
0.02	N/A	N/A	396.5	676.6	978.3	1278.4	1223.8	2576.7	3923.6	1475.4	3120.1	4778.3
0.03	N/A	N/A	437.0	653.6	886.3	1125.5	1208.8	2517.2	3881.6	1410.5	2956.5	4624.0
AT-INPUT: Adversarial Training with Task Loss on the Input Uncertainty												
0.02	0	1.0	219.1	445.2	745.0	1106.1	672.4	1929.0	3334.02	934.4	2706.3	4854.5
		0.5	203.1	438.9	790.1	1246.7	660.7	1932.7	3349.6	914.8	2763.9	4930.5
0.03	0	1.0	239.4	448.6	705.0	1022.1	638.7	1787.5	3220.6	855.5	2445.9	4507.0
		0.5	227.6	556.8	976.6	1435.3	783.7	2074.9	3503.1	1106.8	2948.5	4999.8
AT-PARA: Adversarial Training with Task Loss on the CO Uncertainty												
0	0.05	1.0	212.3	398.84	726.8	1197.4	214.9	590.1	1931.3	399.1	939.1	2620.0
		0.5	206.7	419.3	798.2	1299.9	215.6	741.2	2120.4	419.8	1114.8	3072.6
0	0.15	1.0	214.0	488.7	900.1	1434.5	214.0	221.0	453.2	493.0	925.8	1502.0
		0.5	216.5	423.5	804.6	1309.8	216.5	223.1	465.6	429.6	817.8	1324.6
AT-BOTH: Adversarial Training with Task Loss on the Integrated Uncertainties												
0.02	0.05	1.0	228.2	399.5	655.7	992.0	254.2	823.5	2156.7	400.5	1128.7	2956.2
		0.5	212.1	490.1	879.9	1336.5	289.4	1257.9	2711.8	516.0	1793.9	3903.8
0.03	0.15	1.0	274.7	551.0	856.5	1226.3	274.7	279.0	495.7	550.6	883.4	1274.0
		0.5	239.2	429.1	666.3	977.6	239.2	250.8	508.3	429.0	667.6	1053.7

We first highlight that uncertainties in COs can significantly increase the cost, e.g. 15 times higher than the clean cost when $\epsilon_\phi = 0.15$ for NAT. The AT-MSE performs better for input attacks, compared to the NAT, but performs poorly on the CO attacks. This is because AT-MSE is only trained with input uncertainties. Second, the performance of E2E-AT is similar to conventional adversarial training for image classification [151, 159]. For instance, training with a larger attack budget can result in better robustness on attacks with a smaller budget but can inevitably increase the clean cost. Meanwhile, the hyperparameter α gives a trade-off between clean and robust accuracy in most cases.

In addition to common findings on adversarial robustness of image tasks, some unique findings of E2E-AT are highlighted. First, AT-INPUT and AT-PARA are more effective in the uncertainty with which they are trained. However, it is observed that E2E-AT based on one source of uncertainty can also improve the robustness of the other. For example, in AT-PARA, the cost of the input attack is even lower than that trained by AT-INPUT, when ϵ_x is small. Moreover, AT-PARA is more effective in the integrated attack than AT-INPUT. In fact, any input uncertainty eventually feeds into the COs, which becomes the uncertainties of the predictable parameters in COs. Finally, AT-BOTH not



(a)

(b)

(c)

Figure 4.2: Input space adversarial attack using the exact MILP formulation. (a): Natural Training; (b):Adversarial Training with $\boldsymbol{\delta} = 0.02, \alpha = 1$; (c):Adversarial Training with $\boldsymbol{\delta} = 0.03, \alpha = 1$.

only improves the robustness of integrated uncertainty, but improves the robustness of the individual's. All of the findings demonstrate that the uncertainties of both sources can be treated together in a unified way.

4.6.4 Certified Robustness

Certification on the robustness of the input space is considered, as susceptance $\boldsymbol{b}$ is coupled with the decision variable and Proposition 8 is not applicable. Using Proposition 8 and the mixed integer reformulation of NN (4.4), the exact input space adversarial attack on sample $(\boldsymbol{x}, \boldsymbol{y})$ can be found by a mixed integer linear program (MILP). Interval bound propagation (IBP) [153] is used to estimate the bounds of layers in NN. As for COs, we set $M = 10^5$ by experience. Due to the large computation burden, we randomly sample 15 **same** samples and solve the robust certification using `Gurobi`. A detailed formulation

Table 4.2: CO uncertainties with random sampling of $\boldsymbol{b}$. The adversarial attack reported in Table 4.1 is denoted as PGD-7.

Training Method	**Clean**	$\boldsymbol{\epsilon = 0.05}$		$\boldsymbol{\epsilon = 0.1}$		$\boldsymbol{\epsilon = 0.15}$	
		Random	**PGD-7**	**Random**	**PGD-7**	**Random**	**PGD-7**
NAT	201.6	267.2	754.6	396.6	2044.3	655.5	3468.4
AT-PARA $\boldsymbol{\epsilon_\phi = 0.05, \alpha = 1.0}$	212.3	212.3	214.9	218.4	590.1	313.3	1931.3
AT-PARA $\boldsymbol{\epsilon_\phi = 0.05, \alpha = 0.5}$	206.7	206.7	215.6	222.6	741.2	313.1	2020.4
AT-PARA $\boldsymbol{\epsilon_\phi = 0.15, \alpha = 1.0}$	214.0	214.0	214.0	214.0	221.0	222.9	453.2
AT-PARA $\boldsymbol{\epsilon_\phi = 0.15, \alpha = 0.5}$	216.5	216.5	216.5	216.5	223.1	219.4	465.6

on certified robustness of power system operation can be found in Appendix E.2.

The comparison results of input space adversarial attack can be found in Fig.4.2 in which 'Clean': cost of the clean sample; 'PGD-30': cost of adversarial sample found by 30 PGD steps; 'Exact': cost of adversarial sample found by exact MILP; 'Verify': cost of the adversarial sample by feeding the MILP solutions into the dispatch and redispatch problems. First, the *branch-and-bound* algorithm is applied whose optimality is guaranteed if feasible. Meanwhile, the MILP formulation is verified as the same task-aware cost is achieved when solving the downstream COs parameterized by the optimal attack vector. Second, it can be observed that the exact attack vector causes the worse cost degradation, compared to the PGD-30 attacks. Finally, AT-INPUT can effectively reduce the task-aware cost on the exact input space attack.

4.6.5 Parameter Uncertainties in COs

As shown in (4.19), the uncertainties of CO can be modeled by stochastic CO. To model stochastic susceptance $\boldsymbol{b}$, we randomly alter the susceptance (random attack) for each sample and report the task-aware cost in Table 4.2. Under the random attack, the task-aware cost of NAT increases, clearly demonstrating the need for adversarial training. After E2E-AT, task-aware cost under random attacks is significantly reduced and is equal to the corresponding clean cost when the attack budget is small. Although the clean cost increases, it is still lower than the average cost under random attacks. The task-aware cost is also empirically upper bound by the adversarial attack (e.g. PGD-7), which verifies our argument on connecting the stochastic COs with E2E-AT (4.16) and (4.20).

4.7 Conclusion

This chapter proposes a unified framework for tackling uncertainties in task-aware E2E learning. In is shown that the uncertainties occur at both the input feature of ML and the unpredictable parameter of COs. A robust program is formulated, which is practically solved by adversarial training (E2E-AT). Through theoretical analysis and experiment, it is demonstrated that 1). The unpredictable CO uncertainty can cause significant generalization degradation which has been overlooked before; 2). The optimal adversarial attack on affine-parametric QP can be found by solving the mixed-integer (linear) program; and 3). Adversarial training can effectively improve the robustness of E2E learning in a unified way.

Chapter 5

Task-Aware Machine Unlearning and Its Application in Load Forecasting

In this chapter, the integrated learning and optimization (E2E learning) framework is implemented into a reverse problem, that is, to eliminate the influence of samples from the trained machine learning model with task-aware cost considered. This chapter first introduces the necessity of unlearning consumers' data from both privacy and security perspectives. Then the state-of-the-art literature on existing machine unlearning methods are discussed. Finally, the motivation of this chapter is presented to balance the completeness of unlearning and the task-aware performance of the ML model.

5.1 Introduction

5.1.1 Data Privacy and Security in Load Forecasting

Accurate load forecasting is essential for the security and economic operation of the power system. The deterministic [165] and probabilistic [166] methods are two main categories. Recently, machine and deep learning algorithms have been widely applied to better retrieve spatial and temporal information, which certainly benefit the progression of load forecasting [2]. To fulfill the training purpose, large amount of data from unsecured individuals is collected, which challenges the integrity of data ownership and security.

In power system, the system operator (SO) collects and transfers individual data for various operational purposes. However, this arrangement has raised privacy concerns, as individual load data are sensitive and can be targeted to retrace personal identity and behavior [167]. From the perspective of data security, data collected from unsecured sources are prone to errors and adversaries. For example, the authors of [168] benchmark how poor training data could degrade forecast accuracy by introducing random noise. Furthermore, data poisoning attack is specifically designed to contaminate the training dataset to prevent the load forecaster from being accurate at the test stage [169].

Most of the existing work designs a preventive training algorithm to address concerns about data privacy and security. For example, federated learning is studied, in which each training participant only shares the trained parameters with the central server [170]. In [171], a fully distributed training framework has been proposed in which each participant only shares the parameters with his neighbors. Differential privacy is another privacy-preserving technique used for load forecasting to avoid identifying the individual [172]. To combat poisoning attack, federated learning enhanced with differential privacy is developed in [173]. By weight-clipping and adding noise to the central parameter update, the global model can be resistant to inference attacks to some extent. In addition, gradient quantization is applied, where each participant only uploads the sign of the local gradient [174].

5.1.2 Machine Unlearning

However, training stage prevention is not sufficient when the post-action of removing the impact of those data from the trained forecaster is needed. From privacy concerns, in addition to the right to share the data, many national and regional regulations have certified the consumers' *'right to forget'* [175], such as the European Union's General

Data Protection Regulation (GDPR) and the recent US's California Consumer Privacy Act (CCPA). That is, consumers are eligible to request to destroy their personal records at any stage of the service, including the encoded information in the trained model [176]. Meanwhile, the SO may not be aware of the data defect until the model has been trained and deployed. Obviously, a straightforward approach to removing the impact of part of the dataset is to retrain the model from scratch on the remaining data. However, retraining can be computationally expensive, and an ideal method is to use the already trained model as a starting point.

In this context, machine unlearning (MU) has been introduced in machine learning especially the computer vision community to study the problem of removing a subset of training data, the forget or unlearn dataset, from the trained model. It has recently been extended to other practical fields, such as removing bias in language models [177], unlearning personal information on the Internet of Things [178] and digital twin mobile networks [179], as well as removing malicious samples in wireless communication beam selection problems [180].

Originating from [181] for statistical query learning, MU can be broadly classified into exact and approximate unlearning. Exact unlearnings are developed for specific algorithms, such as k-means [182] and modified random forest [183]. The gradient and the Hessian matrix of the training objective are useful to approximate the influence of samples on the parameters of the trained model. Therefore, the Fisher information [184, 185] and the influence function [186–188] are adopted to unlearn the influence of the forget dataset from the trained model. Motivated by differential privacy, [188] certifies the exactness of data removal in linear classifiers. However, these methods are difficult to generalize to the neural network (NN) [189] with guaranteed unlearning performance. To overcome the problem, a mixed-privacy forgetting is proposed to only unlearn on a linear regression model around the trained NN [187, 188]. Projected gradient unlearning is proposed in [190]. The gradient orthogonal to the column space of gradients of the remaining dataset is adopted to incrementally unlearn the forget dataset without catastrophically forgetting the remaining dataset.

Another line of research assumes that the model is trained with an oracle, that is, by taking into account the future unlearning requirement during training. For example, amnesiac training tracks the contribution of each training batch. When data in a batch is requested to be removed, the batch contribution can simply be subtracted [191]. Alternatively, an exact but efficient retraining algorithm is proposed in [192] in which ensemble models are trained on disjoint subsets of data. Therefore, only the model trained on the unlearned dataset needs to be re-trained. However, when the forget dataset spreads over multiple models, this method becomes less efficient. Finally, the theorem and application of machine unlearning are continually studied and more information can be found in the recent review [193].

5.1.3 Research Gaps

Unlearning Completeness vs Model Performance

Although retraining is usually not a viable option, it is broadly agreed that the *golden rule for unlearning* is to minimize the distance between unlearnt and retrained models [193]. In addition, the unlearning algorithm is *complete* if the unlearnt model is identical to the model re-trained on the remaining dataset. However, we argue that complete unlearning may not be suitable for power system applications. Referring to Table 5.1, when the privacy is mainly concerned (*privacy-driven MU*), although complete unlearning can certainly remove the influence of the forget dataset, it can inevitably degrade the performance of the trained model so that the interest of the remaining customers is harmed [176]. For the *security-driven MU*, the malicious data can still contain useful information. However, the *complete unlearning* not only removes the adverse influence of the forget dataset, but also the useful one.

Therefore, under both privacy and security concerns, machine unlearning is to eliminate the influence of the data from the load forecaster while considering the possible influence on the model performance. This dilemma is modeled as a trade-off between

Table 5.1: The Purposes of Unlearning.

MU Purpose	**Description**	**Target**
Privacy-driven	Some training data contains sensitive information and is asked to remove by the customer.	The **main** target is to unlearn the model as if it is originally trained without the forget data.
Security-driven	Some training data is malicious or biased, whose influence should be removed from the trained model by the SO.	The **main** target is to remove the malicious information from the model while keeping the useful information.

unlearning completeness and *model performance*. How to quantitatively calculate the two factors effectively and efficiently without knowing the re-trained model needs to be investigated.

Physical Meaning of Power System

Apart from the completeness and performance trade-off, directly applying the MU algorithms from machine learning community overlooks the physical meaning of power system. In load forecasting, the ultimate goal is to use the forecast load for downstream tasks, such as dispatching the generator. As shown in [19, 20, 123], the forecast error mismatches the generator cost deviation such that a highly accurate load forecaster may not result in economic power system operation. Intuitively, we argue that the performance of MU also deviates from the accuracy criterion to the task-aware generator cost. Therefore, the cost of generator needs to be evaluated as the performance criterion when dealing with the completeness-performance trade-off.

To our knowledge, this is the first time machine unlearning has been applied to power system applications. Specifically, it is introduced into the load forecasting model (shown in Fig.5.1).

5.2 Machine Unlearning for Load Forecasting

5.2.1 Parametric Load Forecasting Model

In this chapter, we consider the load forecasting problem with n loads/participants. Given dataset $\mathcal{D} = \{(\boldsymbol{x}^i, \boldsymbol{y}^i)\}_{i=1}^N$. Let $\boldsymbol{x}^i \in \mathbb{R}^{n \times M}$ be the feature matrix, i.e., each load has

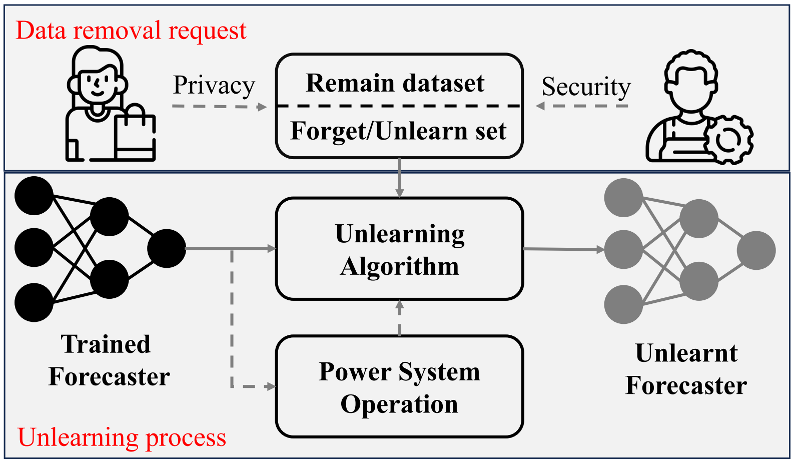


Figure 5.1: A workflow of machine unlearning. The data removal request can be made by privacy and security concerns. An unlearning algorithm is developed to update the forecaster with the role of power system operation being considered.

feature of length M, and $\boldsymbol{y}^i \in \mathbb{R}^n$ be the ground truth load. A parametric model $\boldsymbol{f}(\cdot;\boldsymbol{\theta}):\mathbb{R}^{n\times M}\times\mathbb{R}^P\to\mathbb{R}^n$ can be trained as

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} \ell(\boldsymbol{f}(\boldsymbol{x}^i;\boldsymbol{\theta}),\boldsymbol{y}^i) \tag{5.1}$$

where $\mathcal{L}(\boldsymbol{\theta})$ is the training loss. For simplicity, denote $\ell(\boldsymbol{f}(\boldsymbol{x}^i;\boldsymbol{\theta}),\boldsymbol{y}^i)$ as $\ell^i(\boldsymbol{\theta})$ as the loss on the i-th sample. MSE is commonly used as the training loss by assuming that the forecast error follows Gaussian distribution, i.e., $\ell^i(\boldsymbol{\theta}) = \|\boldsymbol{f}(\boldsymbol{x}^i;\boldsymbol{\theta})-\boldsymbol{y}^i\|_2^2$.

In addition to the training dataset $\mathcal{D}$, there is a test dataset $\mathcal{D}_{\text{test}}$ on which a test criterion can be performed:

$$\mathcal{L}_{\text{test}}(\boldsymbol{\theta}^\star) = \frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}} \ell^i_{\text{test}}(\boldsymbol{\theta}^\star) \tag{5.2}$$

The test criterion $\ell^i_{\text{test}}(\boldsymbol{\theta})$ can be different from the training loss $\ell^i(\boldsymbol{\theta})$. For instance, the load forecasting model can be trained with MSE loss but is usually evaluated by MAPE, etc. In this chapter, we call the loss/criterion such as MSE and MAPE as statistical(-driven) loss/criterion.

5.2.2 Influence Function

The influence function defines a second-order method to evaluate parameter changes when training samples are up-weighted by a small amount [194]. Define a sub-dataset $\mathcal{D}_{\text{up}} \subseteq \mathcal{D}$. For every sample $j \in \mathcal{D}_{\text{up}}$ up-weighted by ϵ^j, the new objective function can be written as

$$\begin{aligned} \boldsymbol{\theta}^\star_{\text{mod}} &= \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{mod}}(\boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} \underbrace{\frac{1}{N}\sum_{i\in\mathcal{D}} \ell^i(\boldsymbol{\theta})}_{\mathcal{L}(\boldsymbol{\theta})} + \underbrace{\frac{1}{N}\sum_{j\in\mathcal{D}_{\text{up}}} \epsilon^j \ell^j(\boldsymbol{\theta})}_{\mathcal{L}_{\text{up}}(\boldsymbol{\theta})} \end{aligned} \tag{5.3}$$

The first-order optimality condition gives that

$$\nabla\mathcal{L}_{\text{mod}}(\boldsymbol{\theta}^\star_{\text{mod}}) = \mathbf{0} \tag{5.4}$$

Apply the first-order Taylor expansion around $\boldsymbol{\theta}^\star$ on (5.4):

$$\nabla\mathcal{L}_{\text{mod}}(\boldsymbol{\theta}^\star) + \nabla^2\mathcal{L}_{\text{mod}}(\boldsymbol{\theta}^\star)(\boldsymbol{\theta}^\star_{\text{mod}} - \boldsymbol{\theta}^\star) \cong \mathbf{0}$$

Consequently, up-weighting samples in $\mathcal{D}_{\text{up}}$ can approximately result in parameter changes

$$\boldsymbol{\theta}^\star_{\text{mod}} - \boldsymbol{\theta}^\star \cong -\left(\nabla^2\mathcal{L}_{\text{mod}}(\boldsymbol{\theta}^\star)\right)^{-1}\nabla\mathcal{L}_{\text{mod}}(\boldsymbol{\theta}^\star) \tag{5.5}$$

Furthermore, since $\nabla\mathcal{L}(\boldsymbol{\theta}^\star) = \mathbf{0}$, $\nabla\mathcal{L}_{\text{mod}}(\boldsymbol{\theta}^\star) = \nabla\mathcal{L}_{\text{up}}(\boldsymbol{\theta}^\star)$. Eq. (5.5) can be rewritten as

$$\boldsymbol{\theta}^\star_{\text{mod}} - \boldsymbol{\theta}^\star \cong -\left(\nabla^2\mathcal{L}_{\text{mod}}(\boldsymbol{\theta}^\star)\right)^{-1}\nabla\mathcal{L}_{\text{up}}(\boldsymbol{\theta}^\star) \tag{5.6}$$

When ϵ_j is small and/or $|\mathcal{D}_{\text{up}}| \ll |\mathcal{D}|$, $\nabla^2\mathcal{L}_{\text{mod}}(\boldsymbol{\theta}^\star) \cong \nabla^2\mathcal{L}(\boldsymbol{\theta}^\star)$. Therefore, (5.6) is further approximated as

$$\boldsymbol{\theta}^\star_{\text{mod}} - \boldsymbol{\theta}^\star \cong -\left(\nabla^2\mathcal{L}(\boldsymbol{\theta}^\star)\right)^{-1}\nabla\mathcal{L}_{\text{up}}(\boldsymbol{\theta}^\star) \tag{5.7}$$

where $\nabla\mathcal{L}_{\text{up}}(\boldsymbol{\theta}^\star) = \frac{1}{N}\sum_{j\in\mathcal{D}_{\text{up}}} \epsilon^j \nabla\ell^j(\boldsymbol{\theta}^\star)$ and $\nabla^2\mathcal{L}(\boldsymbol{\theta}^\star) = \frac{1}{N}\sum_{j\in\mathcal{D}} \nabla^2\ell^j(\boldsymbol{\theta}^\star)$.

We highlight that (5.5)-(5.7) are Newton's update on the parameter with respect to the new objective (5.3). Therefore, for the multivariate linear load forecaster with MSE loss, (5.5) and (5.6) are exact updates on the trained model $\boldsymbol{\theta}^{\star}$.

5.2.3 Machine Unlearning Algorithm

From a data privacy perspective, participants are eligible to ask the SO to remove their data and influence on the trained model $\boldsymbol{\theta}^{\star}$. When a request is made on record j, the corresponding datum $(\boldsymbol{x}^j, \boldsymbol{y}^j)$ needs to be removed from the training dataset. Meanwhile, $\mathcal{D}$ can contain erroneous or malicious data, caused by improper data collection or poisoning attacks, whose influence on the trained forecaster needs to also be removed.

Define $\mathcal{D}_{\text{unlearn}} \subset \mathcal{D}$ as the dataset that needs to be removed and $|\mathcal{D}_{\text{unlearn}}| \ll |\mathcal{D}|$. The remaining dataset is denoted as $\mathcal{D}_{\text{remain}} = \mathcal{D} \setminus \mathcal{D}_{\text{unlearn}}$. A commonly used MU algorithm can be directly derived from the influence function by setting $\epsilon_j = -1$ in (5.3). As a result, (5.5) can be modified as

$$\boldsymbol{\theta}^{\star}_{\text{remain}} \cong \boldsymbol{\theta}^{\star} - \left(\sum_{i \in \mathcal{D}_{\text{remain}}} \nabla^2 \ell^i(\boldsymbol{\theta}^{\star}) \right)^{-1} \sum_{i \in \mathcal{D}_{\text{remain}}} \nabla \ell^i(\boldsymbol{\theta}^{\star}) \tag{5.8}$$

For a linear forecaster, unlearning (5.8) is **complete** as it is guaranteed to converge at $\boldsymbol{\theta}^{\star}_{\text{remain}}$, the model retrained by $\mathcal{D}_{\text{remain}}$. Similar unlearning algorithms can also be derived from (5.6) and (5.7).

5.3 Performance-aware Machine Unlearning

A complete MU algorithm on linear load forecaster such as (5.8) can inevitably influence the performance of the test dataset (will be shown in the simulation). Following the previous work in [195], a performance-aware machine unlearning (PAMU) is derived by re-weighting the remaining samples based on their distinct contribution to the statistic criterion (5.2).

To start, the influence function (5.7) can be further extended to assess the performance change of the test set due to the up-weighted objective (5.3) [189, 196]. The performance on the test dataset for model parameterized by $\boldsymbol{\theta}^{\star}_{\text{remain}}$ can be written as

$$\mathcal{L}_{\text{test}}(\boldsymbol{\theta}^{\star}_{\text{remain}}) = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \ell^{i}_{\text{test}}(\boldsymbol{\theta}^{\star}_{\text{remain}}) \tag{5.9}$$

Applying first-order Taylor expansion on (5.9) gives:

$$\mathcal{L}_{\text{test}}(\boldsymbol{\theta}^{\star}_{\text{remain}}) \cong \underbrace{\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \ell^{i}_{\text{test}}(\boldsymbol{\theta}^{\star})}_{\mathcal{L}_{\text{test}}(\boldsymbol{\theta}^{\star})} + \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \nabla \ell^{i}_{\text{test}}(\boldsymbol{\theta}^{\star})^{T} (\boldsymbol{\theta}^{\star}_{\text{remain}} - \boldsymbol{\theta}^{\star}) \tag{5.10}$$

To eliminate the performance change (5.10), the remaining dataset can be re-weighted. The idea is straightforwardly that, after unlearning, different remaining samples will have different influence on the performance, which needs to be re-weighted as if they are being re-trained.

The new objective function on the re-weighted remaining dataset can be written as

$$\boldsymbol{\theta}^{\star}_{\text{remain},\epsilon} = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i \in \mathcal{D}_{\text{remain}}} \epsilon^{i} \ell^{i}(\boldsymbol{\theta}) \tag{5.11}$$

where ϵ^{i} is an unknown weight for sample i in the remaining dataset. Referring to (5.5), the parameter changes can be approximated as

$$\boldsymbol{\theta}^{\star}_{\text{remain},\epsilon} - \boldsymbol{\theta}^{\star} \cong - \left(\sum_{i \in \mathcal{D}_{\text{remain}}} \epsilon^{i} \nabla^{2} \ell^{i}(\boldsymbol{\theta}^{\star}) \right)^{-1} \sum_{i \in \mathcal{D}_{\text{remain}}} \epsilon^{i} \nabla \ell^{i}(\boldsymbol{\theta}^{\star}) \tag{5.12}$$

Plugging (5.12) into (5.10), the performance changes can be written as:

$$\mathcal{L}_{\text{test}}(\boldsymbol{\theta}^{\star}_{\text{remain},\epsilon}) - \mathcal{L}_{\text{test}}(\boldsymbol{\theta}^{\star}) \cong \boldsymbol{m}^{T} \sum_{i \in \mathcal{D}_{\text{remain}}} \epsilon^{i} \nabla \ell^{i}(\boldsymbol{\theta}^{\star}) \tag{5.13}$$

where

$$\boldsymbol{m}^{T} = - \frac{1}{N_{\text{test}}} \sum_{i \in \mathcal{D}_{\text{test}}} \nabla \ell^{i}_{\text{test}}(\boldsymbol{\theta}^{\star})^{T} \left(\sum_{i \in \mathcal{D}_{\text{remain}}} \epsilon^{i} \nabla^{2} \ell^{i}(\boldsymbol{\theta}^{\star}) \right)^{-1} \tag{5.14}$$

When ϵ^i is close to 1, the $\boldsymbol{m}$ vector can be approximated as

$$\tilde{\boldsymbol{m}}^T = -\frac{1}{N_{\text{test}}} \sum_{i \in \mathcal{D}_{\text{test}}} \nabla \ell^i_{\text{test}}(\boldsymbol{\theta}^\star)^T \left(\sum_{i \in \mathcal{D}_{\text{remain}}} \nabla^2 \ell^i(\boldsymbol{\theta}^\star) \right)^{-1} \tag{5.15}$$

Our goal is to find an optimal weights to improve the test set performance, which can be formulated as a constrained optimization problem:

$$\begin{aligned} \boldsymbol{\epsilon}^\star = \arg\min_{\boldsymbol{\epsilon}} \quad & \tilde{\boldsymbol{m}}^T \sum_{i \in \mathcal{D}_{\text{remain}}} \epsilon^i \nabla \ell^i(\boldsymbol{\theta}^\star) \\ \text{subject to} \quad & \frac{1}{N_{\text{remain}}} \|\boldsymbol{\epsilon} - \mathbf{1}\|_1 \leq \lambda_1, \quad \|\boldsymbol{\epsilon} - \mathbf{1}\|_\infty \leq \lambda_\infty \end{aligned} \tag{5.16}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^{|\mathcal{D}_{\text{remain}}|}$ and $\mathbf{1} \in \mathbb{R}^{|\mathcal{D}_{\text{remain}}|}$.

In (5.16), the weights of the remaining samples are optimized so that the influence of forgetting $\mathcal{D}_{\text{unlearn}}$ is reduced. Since the first-order Taylor expansion (5.10) is a local approximation, the 1-norm and inf-norm constraints are added to control aggregated and individual re-weighting. When $\lambda_1 \to 0$ or $\lambda_\infty \to 0$, $\boldsymbol{\epsilon} \to 1$, representing complete machine unlearning (5.8). When λ_1 and λ_∞ become larger, the performance of the test dataset improves, while the completeness of the unlearning is reduced. Therefore, by controlling λ_1 and λ_∞, the trade-off between MU completeness and performance changes can be balanced.

Since both $\tilde{\boldsymbol{m}}$ and $\nabla \ell^i(\boldsymbol{\theta})^\star, i \in \mathcal{D}_{\text{remain}}$ are calculated in advance, (5.16) is a convex optimization problem that can be easily solved. Once the optimal weights $\boldsymbol{\epsilon}^\star$ are optimized, we can unlearn $\mathcal{D}_{\text{unlearn}}$ through (5.12). Regarding different choices of ℓ_{test}, e.g. MSE and MAPE, the remaining dataset can be reweighted in distinct manners. It is also possible to integrate different criteria.

Furthermore, compared to [195], the objective of (5.16) does not take the absolute value. Therefore, the objective of (5.16) can be negative and it is allowed to improve performance beyond the originally trained model. Any uncovered biased data in $\mathcal{D}_{\text{remain}}$ will be assigned a smaller weight, and unlearning becomes a one-step continual learning on the re-weighted samples. Re-weighting the samples to improve the model performance has been used in other load forecasting algorithm [197]. However, we directly find the suitable weights on the trained parameter $\boldsymbol{\theta}^{\star}$ through an optimization problem (5.16) and update the model using the second-order approach (5.12).

5.4 Task-aware Machine Unlearning

5.4.1 Formulation and Algorithm

In power systems, the forecast load is further used to schedule generators, and the statistic accuracy of the forecast load is eventually converted into the deviation of the generator cost, which is strongly linked to the value of each sample as well as the profit of the SO and participants. As a result, PAMU guided by the statistic-driven criterion may not reflect on the ultimate goal of power system operation, and a further step on PAMU is needed to balance the generator cost, which can be done by taking the generator cost as the new test criterion.

To measure the impact of model parameter $\boldsymbol{\theta}$ on the objective (5.18), the following task-aware criterion $\mathcal{L}_{\text{gen}}(\boldsymbol{\theta})$ can be formulated:

$$
\begin{aligned}
&\min_{\boldsymbol{\theta}} \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \ell_{\text{gen}}^{i}(\boldsymbol{\theta}) \\
&\text{s.t.} \begin{cases}
\text{(Re-dispatch):} \\
(\boldsymbol{P}_{ls}^{i\star}, \boldsymbol{P}_{gs}^{i\star}) \in \arg\min\{c_{ls2}\|\boldsymbol{P}_{ls}^{i}\|_2^2 + c_{gs2}\|\boldsymbol{P}_{gs}^{i}\|_2^2 + c_{ls}\|\boldsymbol{P}_{ls}^{i}\|_1 + c_{gs}\|\boldsymbol{P}_{gs}^{i}\|_1 : \\
\qquad\qquad (\boldsymbol{P}_{ls}^{i}, \boldsymbol{P}_{gs}^{i}) \in \mathcal{C}_{\text{redispatch}}(\boldsymbol{P}_{g}^{i\star}, \boldsymbol{y}^{i})\} \\
\text{(Dispatch):} \\
\boldsymbol{P}_{g}^{i\star} \in \arg\min\{\boldsymbol{P}_{g}^{iT}\boldsymbol{Q}_{g}\boldsymbol{P}_{g}^{i} + \boldsymbol{c}_{g}^{T}\boldsymbol{P}_{g}^{i} + c_{ls}\|\boldsymbol{s}^{i}\|_1 : (\boldsymbol{P}_{g}^{i}, \boldsymbol{s}^{i}) \in \mathcal{C}_{\text{dispatch}}(\hat{\boldsymbol{y}}^{i})\} \\
\text{(Forecast):} \\
\hat{\boldsymbol{y}}^{i} = \boldsymbol{f}(\boldsymbol{x}^{i}; \boldsymbol{\theta})
\end{cases} \\
&\text{for all } i = 1, \cdots, |\mathcal{D}_{\text{test}}|
\end{aligned}
\tag{5.17}
$$

Detailed formulations can be found in Appendix F.1. The task-aware criterion (5.17) can be viewed as a trilevel optimization problem with two lower levels, taking the expectation over the test dataset. For each sample, the lower level one is a dispatch problem that minimizes the generator cost subject to the system operation constraint $\mathcal{C}_{\text{dispatch}}$. $\boldsymbol{P}_g$ is the generator dispatch. Lower level two is a re-dispatch problem which aims to balance any under- or over-generation due to inaccurate forecast through load shedding $\boldsymbol{P}_{ls}$ and generation storage $\boldsymbol{P}_{gs}$, under the constraint set $\mathcal{C}_{\text{redispatch}}$. The upper level, which represents the expected operation cost, can be determined as the integration of the two stages:

$$\begin{aligned} & \ell_{\text{gen}}(\boldsymbol{P}_g, \boldsymbol{P}_{ls}, \boldsymbol{P}_{gs}; \boldsymbol{\theta}) \\ = \quad & \boldsymbol{P}_g^T \boldsymbol{Q}_g \boldsymbol{P}_g + c_{ls2}\|\boldsymbol{P}_{ls}\|_2^2 + c_{gs2}\|\boldsymbol{P}_{gs}\|_2^2 + \boldsymbol{c}_g^T \boldsymbol{P}_g + c_{ls1}\|\boldsymbol{P}_{ls}\|_1 + c_{gs1}\|\boldsymbol{P}_{gs}\|_1 \end{aligned} \tag{5.18}$$

When $\boldsymbol{\theta}$ is fixed and if each lower-level problem has a unique optimum, (5.17) is the expected real-time power system operation cost on the test dataset.

Referring to (5.10), to evaluate the influence on the generator cost, the gradient $\nabla \ell^i_{\text{gen}}(\boldsymbol{\theta}^\star)$ needs to be calculated, which seems to be a problem due to the nested structure and constraints in (5.17). To solve the problem, firstly, for each sample i, it can be observed that the lower level problems are sequentially connected, that is, the input to stage one problem is the forecast load while the input to stage two problem is the generator dispatch status from stage one. Second, the lower-level problems are also independent among samples and the constraints. Therefore, the lower-level optimizations can be viewed as composite function for each sample. Let $\boldsymbol{P}^i_g = \boldsymbol{p}^\star_1(\hat{\boldsymbol{y}}^i)$ and $\boldsymbol{P}^i_{ls,gs} = \boldsymbol{p}^\star_2(\boldsymbol{P}^i_g, \boldsymbol{y}^i)$ be the optimal solution map for dispatch and re-dispatch, the individual generator cost (5.18) can be written as a composite function:

$$\ell^i_{\text{gen}}(\boldsymbol{\theta}^\star) = \ell^i_{\text{gen}}(\boldsymbol{p}^\star_1(\hat{\boldsymbol{y}}^i), \boldsymbol{p}^\star_2(\boldsymbol{p}^\star_1(\hat{\boldsymbol{y}}^i), \boldsymbol{y}^i)) \tag{5.19}$$

Alternatively, we can view the lower-level optimizations as sequential layers upon the parametric forecasting model. The layer, which represents a constrained optimization

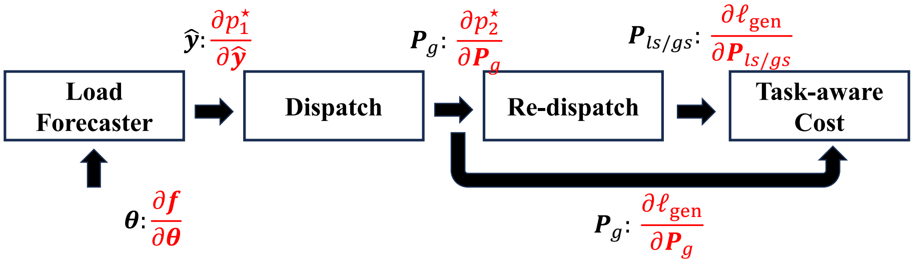

$\hat{\boldsymbol{y}}$: $\frac{\partial p_1^\star}{\partial \hat{\boldsymbol{y}}}$
$\boldsymbol{P}_g$: $\frac{\partial p_2^\star}{\partial \boldsymbol{P}_g}$
$\boldsymbol{P}_{ls/gs}$: $\frac{\partial \ell_{\text{gen}}}{\partial \boldsymbol{P}_{ls/gs}}$
Load Forecaster
Dispatch
Re-dispatch
Task-aware Cost
$\boldsymbol{\theta}$: $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}$
$\boldsymbol{P}_g$: $\frac{\partial \ell_{\text{gen}}}{\partial \boldsymbol{P}_g}$

Figure 5.2: The structure of tri-level optimization (5.17) viewed as layers in the forward pass. The gradients used in (5.20) are highlighted in red.

problem, is named as differentiable convex layer [106].

Consequently, TAMU can be achieved by replacing the statistic metric $\ell^i_{\text{test}}(\boldsymbol{\theta}^\star)$ by $\ell^i_{\text{gen}}(\boldsymbol{\theta}^\star)$, followed by finding the weights of the remaining dataset (5.16) and updating the parameters by (5.12).

The last issue that needs to be resolved is to calculate the gradient of (5.19) as required by (5.10). From the chain rule, the gradient of (5.19) can be written as:

$$\begin{aligned}&\frac{\partial \ell^i_{\text{gen}}(\boldsymbol{\theta}^\star)}{\partial \boldsymbol{\theta}}\\ =& \left(\frac{\partial \ell^i_{\text{gen}}(\boldsymbol{P}^i_g, \boldsymbol{P}^i_{ls,gs})}{\partial \boldsymbol{P}_g} + \frac{\partial \ell^i_{\text{gen}}(\boldsymbol{P}^i_g, \boldsymbol{P}^i_{ls,gs})}{\partial \boldsymbol{P}_{ls,gs}} \frac{\partial \boldsymbol{p}^\star_2(\boldsymbol{P}^i_{\boldsymbol{g}})}{\partial \boldsymbol{P}_g}\right) \times \frac{\partial \boldsymbol{p}^\star_1(\hat{\boldsymbol{y}}^i)}{\partial \hat{\boldsymbol{y}}} \frac{\partial \boldsymbol{f}(\boldsymbol{\theta}^\star)}{\partial \boldsymbol{\theta}}\end{aligned} \tag{5.20}$$

with the gradient flow highlighted in Fig.5.2. In (5.20), the gradient $\partial \ell^i_{\text{gen}}(\boldsymbol{\theta}^\star)/\partial \boldsymbol{\theta}$ exists if the gradients through the differentiable convex layers, namely $\partial \boldsymbol{p}^\star_1(\hat{\boldsymbol{y}}^i)/\partial \hat{\boldsymbol{y}}$ and $\partial \boldsymbol{p}^\star_2(\boldsymbol{P}^i_{\boldsymbol{g}})/\partial \boldsymbol{P}_g$, exist, which is fulfilled under some assumptions in the following proposition.

Proposition 9. *The gradients $\partial \boldsymbol{p}^\star_1(\hat{\boldsymbol{y}}^i)/\partial \hat{\boldsymbol{y}}$ and $\partial \boldsymbol{p}^\star_2(\boldsymbol{P}^i_{\boldsymbol{g}})/\partial \boldsymbol{P}_g$ exist, which do not depend on $\hat{\boldsymbol{y}}^i$ and $\boldsymbol{P}^i_{\boldsymbol{g}}$, respectively, if 1). $\boldsymbol{Q}$ is positive definite, $\mathbf{c}_{ls2}$ and $\mathbf{c}_{gs2}$ are positive; and 2). The linear independent constraint qualification (LICQ) is satisfied at the optimum of each of the lower-level problems.*

The proof can be found in Appendix B.10.

5.4.2 Extension to Neural Network based Load Forecaster

The unlearning algorithm (5.8) is complete on the linear load forecaster as the training objective is quadratic. However, this condition is usually not satisfied for neural networks. In the meantime, its Hessian can be singular due to early stop of training. This makes the influence function approximate poorly to the parameter and performance changes [189], and it becomes harder to evaluate the trade-off between unlearning completeness and model performance in PAMU and TAMU.

To address this problem, we assume that there exists a load forecasting model which is not trained by the consumers' data in the service provided by SO. The model can be a pre-trained model which is publicly available or can be trained on historic non-sensitive data by the SO. Using the idea of transfer learning [198], the SO can then use the pre-trained load forecaster as a deep feature extractor and use the consumers' data to fine tune the last layer. Therefore, only the last layer needs to be unlearnt.

Practically, we first divide the training dataset into pre-trained and user-sensitive data as $\mathcal{D}_{\text{pre}}$ and $\mathcal{D}_{\text{sen}}$, respectively, with $\mathcal{D}_{\text{pre}} \cap \mathcal{D}_{\text{sen}} = \emptyset$. The pre-trained data is assumed to be collected neutrally, which does not violate any participant's privacy and is error-free, while the user-sensitive data may not be. We then pre-train a load forecaster on the $\mathcal{D}_{\text{pre}}$ using regular stochastic gradient descent (SGD), and the trained model (except for the last layer) can be used as a deep feature extractor $\boldsymbol{f}(\cdot;\boldsymbol{\theta}^{\star}_{\text{FE}})$. As illustrated in Fig.5.3, $\mathcal{D}_{\text{sen}}$ is further used to fine-tune Linear Layer 2 by the MSE loss. Since using a stochastic gradient method can introduce uncertainties, we propose to fine-tune the last layer analytically on $\mathcal{D}_{\text{sen}}$ according to the following proposition.

Proposition 10. *The optimization problem of minimizing the MSE loss on a linear layer without activations is quadratic and has unique minimizer if the extracted features from* $\boldsymbol{f}(\cdot;\boldsymbol{\theta}^{\star}_{FE})$ *are linearly independent.*

The proof can be found in Appendix B.11.

According to Proposition 10, `ReLU` activation cannot be used in Linear Layer 1 as it can result in trivial output when the extracted features are negative for some of the samples in

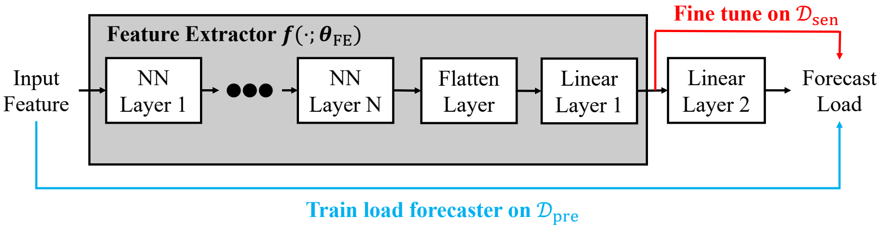

Feature Extractor $f(\cdot;\theta_{\text{FE}})$
Fine tune on $\mathcal{D}_{\text{sen}}$
Input Feature
NN Layer 1
NN Layer N
Flatten Layer
Linear Layer 1
Linear Layer 2
Forecast Load
Train load forecaster on $\mathcal{D}_{\text{pre}}$

Figure 5.3: Structure of NN based load forecaster and feature extractor. All the layers except for the Flatten Layer and the Linear Layer 2 contain activations.

the remaining dataset. When an unlearning is requested, the same unlearning algorithms developed previously can be applied on Linear Layer 2 alone and MU (5.8) is complete. Since the feature extractor trains only on the pre-train data, it does not contain sensitive information that needs to be unlearnt.

5.4.3 Computations

In this section, we discuss some computational issues and some useful open-source packages for the developed unlearning algorithms.

Inversion of Hessian

Machine unlearning (5.8) and calculation of vector $\tilde{\boldsymbol{m}}$ in PAMU and TAMU require matrix inversion of the Hessian matrix. In general, second-order differentiation on training loss is time consuming, as storing and inverting the Hessian matrix requires $\mathcal{O}(d^3)$ operations, where d represents the number of parameters in the load forecast model.

Using $\tilde{\boldsymbol{m}}$ (5.15) as a example:

$$\tilde{\boldsymbol{m}}^T = \underbrace{-\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}}\nabla\ell^i_{\text{test}}(\boldsymbol{\theta}^\star)^T}_{\boldsymbol{v}^T\in\mathbb{R}^{1\times d}}\underbrace{\left(\sum_{i\in\mathcal{D}_{\text{remain}}}\nabla^2\ell^i(\boldsymbol{\theta}^\star)\right)^{-1}}_{\boldsymbol{H}^{-1}\in\mathbb{R}^{d\times d}} \tag{5.21}$$

Calculating $\tilde{\boldsymbol{m}}$ can be reduced to solve a linear system:

$$\boldsymbol{H}\cdot\tilde{\boldsymbol{m}} = \boldsymbol{v} \tag{5.22}$$

The conjugate gradient (CG) descent algorithm can be applied to solve (5.22) up to d iterations. We also apply the Hessian vector product (HVP) [199] to directly calculate $\boldsymbol{H}\tilde{\boldsymbol{m}}^k$ for the k-th iteration in CG so that the Hessian matrix will never be explicitly calculated and stored. HVP is computationally efficient as it only requires one modified forward and backward pass. Similarly, to implement PAMU or TAMU, we can modify the objective directly into the sum of training loss weighted by $\boldsymbol{\epsilon}^\star$ from (5.16) and implement the same CG and HVP procedure. We implement these functionalities using a modified version of `Torch-Influence` package [189].

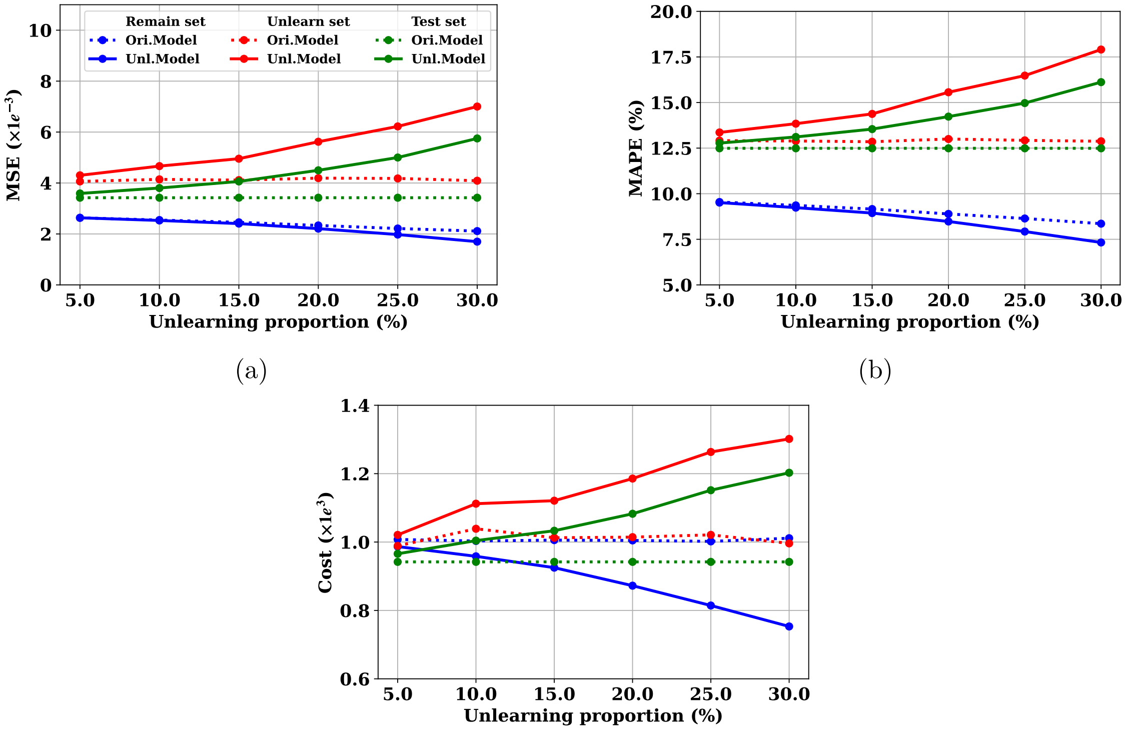
Differentiable Convex Layer

In TAMU, the gradient of generator cost (5.20) can be analytically written according to Proposition 13 in Appendix B.10. It also requires the forward pass to solve the dispatch and re-dispatch problems. In the simulation, we model the operation problems and (5.16) by `Cvxpy` [148]. When calculating the gradient, we use `PyTorch` automatic differentiation package and `CvxpyLayers` [106] to implement fast batched forward and backward passes.

5.5 Experiments and Results

5.5.1 Simulation Settings

We used an open-source dataset from the Texas Backbone Power System [164] which includes meteorological and calendar features and loads in 2019 with a resolution of one hour. The dispatch and re-dispatch problems are solved on a modified IEEE bus-14 system to demonstrate the proposed algorithms. Three parametric load forecasting models, namely multivariate linear regression, convolutional neural network (CNN), and MLP-Mixer [200], are trained by MSE loss. Detailed experimental settings can be found in Appendix F.2.



(a)

(b)

(c)

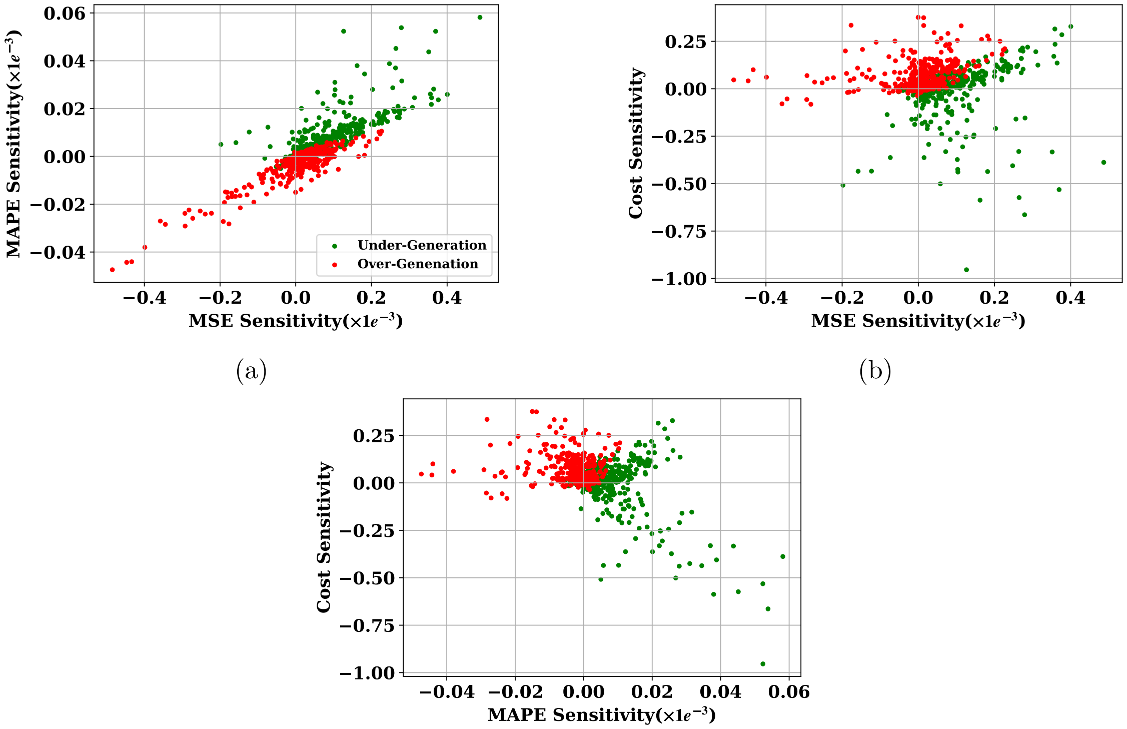
Figure 5.4: Performance of complete machine unlearning algorithm (5.8) on remain (blue), unlearn (red) and test dataset (blue) of the **linear** load forecaster. The dotted curves report the performance of the original model and the solid curves are the performance of the unlearnt model. Different unlearning criteria are used with (a): MSE; (b): MAPE; (c): Cost.

5.5.2 Unlearning Performance on the Linear Model

Unlearning Performance

Unlearning performances on the linear load forecasting model under various unlearning criteria are summarized in Fig.5.4. We have verified that the unlearning algorithm (5.8) results in the same updated parameter as the one re-trained on the remaining dataset under all unlearning rates.

Note that the dotted curves, which represent the performance of the original model, only slightly change over the various unlearning ratios. Broadly speaking, the performance gaps between the unlearnt and original models becomes larger as the unlearning ratio increases. Especially, all the performance criteria on the test dataset become worse



(a)

(b)

(c)

Figure 5.5: Relationship on the influences of MSE, MAPE, and Cost criteria on the test dataset from the samples in remain dataset. The r values are Pearson correlation coefficients. (a): MSE and MAPE ($\boldsymbol{r = 0.829}$); (b): MSE and Cost ($\boldsymbol{r = 0.073}$); (c): MAPE and Cost ($\boldsymbol{r = -0.480}$)

when the unlearning proportion increases, which verifies the statement that unlearning can inevitably degrade the generalization ability of the trained model. For instance, the generator cost can increase by 20% when 20% of the training data are unlearnt. In contrast, the performance of the remain dataset improves as the unlearning ratio increases. This is because when the original model is unlearnt, the model parameters are updated and fitted more on the remaining dataset. Moreover, it can be observed that the trends of performance changes of the unlearnt model are distinct for different criterion. In detail, the generator cost (Fig.5.4c) diverges more significantly from the original model, compared to MSE and MAPE.

Performance Sensitivity Analysis

For each sample in the remaining dataset, we can calculate its influence on the expected performance of the test dataset. The remaining dataset is chosen as it is re-weighted by PAMU and TAMU. For $i \in \mathcal{D}_{\text{remain}}$, the influence can be found by (5.13) and (5.15), i.e.,

$$\mathcal{I}^i_{\text{test}} - \frac{1}{N_{\text{test}}} \sum_{j \in \mathcal{D}_{\text{test}}} \nabla \ell^j_{\text{test}}(\boldsymbol{\theta}^\star)^T \left(\sum_{j \in \mathcal{D}_{\text{remain}}} \nabla^2 \ell^j(\boldsymbol{\theta}^\star) \right)^{-1} \nabla \ell^i(\boldsymbol{\theta}^\star) \tag{5.23}$$

where the test loss $\ell_{\text{test}}(\cdot)$ can be MSE, MAPE or Cost (5.19). To visualize the relationship among these criteria, we randomly draw 1k samples with equal size of under- and over-generation cases from the remaining dataset. For each sample, the under-generation means that the sum of the forecast loads is lower than the sum of the ground-truth load, and the over-generation is opposite. The relationships of any two of the criteria are illustrated in Fig.5.5 with associated Pearson correlation coefficients (the r value) calculated. Using MSE and MAPE as an example, the Pearson correlation coefficient is defined as

$$r_{\text{MSE,MAPE}} = \frac{\sum_i (\mathcal{I}^i_{\text{MSE}} - \bar{\mathcal{I}}_{\text{MSE}})(\mathcal{I}^i_{\text{MAPE}} - \bar{\mathcal{I}}_{\text{MAPE}})}{\sqrt{\sum_i (\mathcal{I}^i_{\text{MSE}} - \bar{\mathcal{I}}_{\text{MSE}})^2} \sqrt{\sum_i (\mathcal{I}^i_{\text{MAPE}} - \bar{\mathcal{I}}_{\text{MAPE}})^2}} \tag{5.24}$$

where $\bar{\mathcal{I}}_{\text{MSE}}$ and $\bar{\mathcal{I}}_{\text{MAPE}}$ are the mean of MSE and MAPE influence, respectively.

Since the performance changes are modeled linearly by first-order Taylor expansion (5.10) and the objective of re-weighting optimization is also linear (5.16), Pearson correlation coefficient is a suitable indicator of the linear relationship. In Fig.5.5, positive sensitivity represents the degradation of performance after unlearning such sample. That is, after this sample is unlearnt, the MSE, MAPE, or average generator cost on test dataset increases.

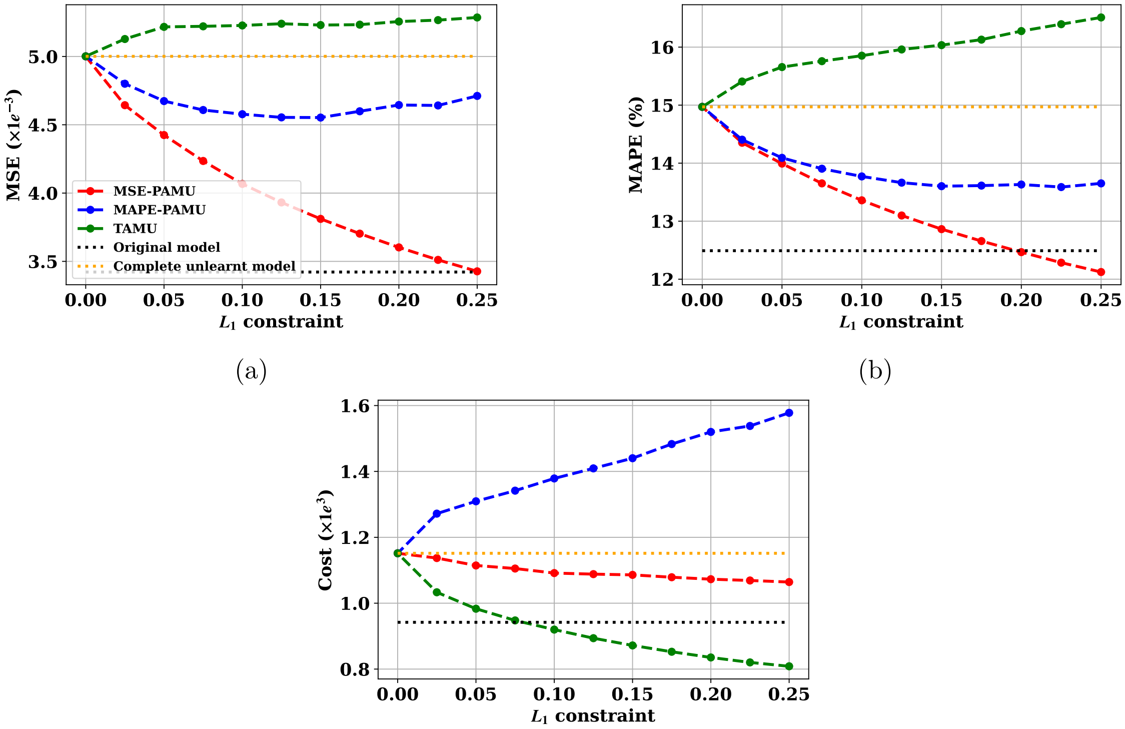
First, the Pearson correlation coefficients have clearly demonstrated that there exists a strong positive linear relationship between the two statistic criteria (0.829), while this relationship is insignificant between the statistic and task-aware criteria (0.073 between MSE and Cost and -0.480 between MAPE and Cost). These distinct relationships imply that balancing performance by one statistical criterion is likely effective on the other. In contrast, balancing the performance by statistical criteria can unlikely be effective on the generator cost and vice versa. Secondly, as the under-generation is more costly than the over-generation, unlearning an under-generation sample tends to reduce the overall generator cost with negative sensitivities. As shown by Fig.5.5b and Fig.5.5c, if the sensitivities are projected to the y-axis, most of the negative sensitivities are contributed by the under-generation samples, which verifies our intuition. However, it does not occur in MSE and MAPE as they are almost centrally symmetric around the origin in Fig.5.5a.

The above discussions can verify the intuition that the statistic performance cannot reflect and may even conflict with the task-aware operation cost.

Performances of PAMU and TAMU

The performance of PAMU and TAMU on the test dataset is reported in Fig.5.6 in which 25% training data is removed. To balance the trade-off, λ_1 is varied and the inf-norm constraint λ_∞ in (5.16) is set as 1. That is, the weight of a remaining sample can very from 0 to 2. First, unlearning by balancing one of the criteria can effectively maintain the performance of the same criterion (e.g., red curve in Fig.5.6a, blue curve in Fig.5.6b, and green curve in Fig.5.6c). When λ_1 approaches 0, the PAMU and TAMU become complete with the same performance as the retrained model in all criteria, as no samples can be re-weighted. When λ_1 increases, the performance of the original model is recovered and the divergence to the retrained model increases. After λ_1 is further increased, better performance is achieved, resulting in a new type of continual learning through sample re-weighting. As a result, the proposed PAMU and TAMU can effectively balance the completeness and performance trade-off in MU by changing λ_1. In addition, Fig.5.7 illustrates the parameter difference to the retrained model (evaluated by 2-norm) vs the generator cost, which clearly demonstrates the trade-off as well.

Meanwhile, it is observed that the cost curves perform differently compared to the MSE and MAPE curves. When balancing the cost, both MSE and MAPE get worse. In contrast,



(a)

(b)

(c)

Figure 5.6: Performance of PAMU and TAMU with different test criteria. a), b), and c) are performances on the test dataset evaluated by MSE, MAPE, and average generator cost, respectively. The performance of the original model and the model unlearnt by complete unlearning (5.8) are represented by the black and orange lines, respectively

balancing the MSE can also keep/improve the MAPE performance to some extent, and visa versa. This observation is in line with the analysis on the Pearson correlation coefficient in the previous section.

5.5.3 Unlearning Performance on the NN Forecaster

Since the fine-tuning objective on $\mathcal{D}_{\text{sen}}$ is quadratic (Proposition 10), the direct unlearning is also complete for the NN load forecaster. We can expect that the unlearning behaviors are similar to the linear counterpart. Therefore, we only highlight some of the simulation results and leave details in Appendix F.3. Similarly to the linear counterpart, unlearning part of the training dataset can deteriorate the performance of the test set. The performance-aware unlearning algorithm can effectively balance the unlearning complete-

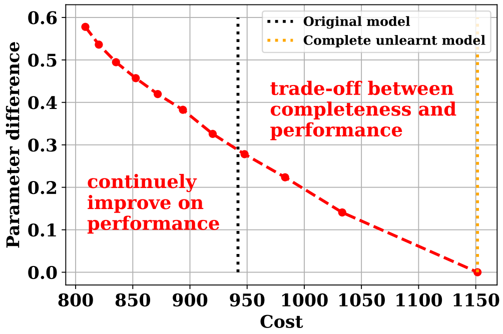


Figure 5.7: Trade-off between MU completeness and the operation cost.

ness and model performance, which is more effective on the criterion it is evaluated on. However, instead of having opposite statistical and cost trends in Fig.5.6, all criteria are improved with decay speed. This is because the NN-based load forecaster is more accurate than the linear counterpart, resulting in a less significant misalignment between the load forecast accuracy and the generator cost.

5.6 Conclusion

This chapter introduces machine unlearning algorithm for load forecasting model to eliminate the influence of data that is adversarial or contains sensitive information of individuals. The contributions of this chapter are summarized as follows.

- **Machine Unlearning**: The influence of forget dataset on the trained model is evaluated using the influence function-based approach, which is eliminated by Newton's update.

- **Completeness-Performance Trade-off**: Complete unlearning is shown to inevitably influence the statistical performance of the load forecaster, such as MSE and MAPE. To overcome the dilemma, the influence function is used to quantify the impact on the statistical performance of each sample, allowing the remaining dataset to be reweighted through optimization and the performance to be improved through performance-aware machine unlearning (PAMU).

- **Task-aware Machine Unlearning**: Finally, statistical performance has been shown to not reflect the ultimate goal of power system operation, such as minimizing the cost of generator dispatch. Therefore, a task-aware machine unlearning (TAMU) is proposed by formulating the unlearning objective as a trilevel optimization. The existence of the gradient of such task-aware objective is theoretically proved, which is key to sample re-weighting. The simulation results verify that the proposed task-aware algorithm can significantly reduce the generator cost on the test dataset by compensating for the unlearning completeness.

Chapter 6

Conclusion and Future Work

6.1 Conclusions

This thesis delves into the challenge of integrating data and the model for secure and robust decision-making, with a specific focus on power system operations. Here, the term “data” encompasses data-driven decision-making methodologies such as deep learning algorithm, while “model” refers to model-based decision-making techniques such as optimization programming. We argue that the co-design of data and model is a crucial stride towards the evolution of automated decision-making in the future.

Given that real-time tasks often entail multi-step decision-making processes, data-driven operations can be classified into a purely data-driven approach and a mixed data and model-based approach. The latter can be further subcategorized into separate, sequential, and integrated blending of data and model, with concise definitions and properties provided in the thesis’s Introduction. Two key theoretical frameworks, specifically the **sequential** and **integrated** formulations, are technically highlighted. Their effectiveness is demonstrated through applications in the detection of FDI attacks and objective-based load forecasting, respectively.

In general, the advantages of blending data and model include:

- In the context of data-driven algorithms like deep learning, incorporating physical information during both the training and inference stages can markedly enhance their feasibility and reliability. In the case of a data-driven attack detector, a subsequent model-based verification can statistically manage the FPR. Similarly, for a data-driven attack identifier, integrating physics-informed algorithms allows encoding prior knowledge of the attack to improve identification accuracy. In E2E learning, the inclusion of a differentiable convex layer serves to guide the parametric space of the data-driven model towards the objectives of downstream tasks.

- In the realm of model-based optimization, uncertain parameters can be learned and quantified through a data-driven algorithm. This approach enables the shift of the online computation burden into the offline training phase. Additionally, for robust and stochastic optimization, the uncertainty set and the underlying distribution of parameters can be intelligently learned, with the purpose of maximizing the profit of decision-making.

In particular, the contribution of each chapter is given as follows.

Part I of the thesis focuses on sequential design of the learning and optimization algorithm, with application to detecting FDI attacks on power system state estimation. To start, this thesis first proposes a new model-based robust MTD algorithm in Chapter 2. We demonstrate that the existing criterion for MTD effectiveness based on the rank condition on the composite pre- and post-MTD measurement matrix is insufficient when the measurement is noisy. Therefore, it is theoretically proved that the weakest point for any given MTD strategy corresponds to the smallest principal angle between the pre- and post-MTD measurement Jacobian subspaces, and the worst-case detection rate is proportional to the sine of this angle, with the impact of measurement noise being explicitly considered. For MTDs with a complete configuration, an optimization problem is formulated to maximize the minimum principal angle. It is then demonstrated that the worst-case detection rate of the grid with incomplete configuration cannot be improved. Therefore, an iterative algorithm is formulated to maximize the minimal non-zero principal angle while limiting the chance of attacking on the subspace that cannot be detected.

The robust MTD can be conceptualized as a robust optimization problem with an attack uncertainty set equivalent to the entire feasible measurement space. Chapter 3 demonstrates an extension to robust MTD and highlights the advantages of the sequential formulation. The proposed data-driven event-triggered MTD comprises two components. Firstly, an unsupervised LSTM-AE detector is trained on the historical normal measurement dataset. Whenever the data-driven detector triggers an alarm, an attack identification algorithm is employed to project the attacked measurement onto the manifold defined by the LSTM-AE and the feasible set defined by the power flow equation. The resulting permissible attack set is constructed and utilized as the uncertainty set for robust MTD. As an extension to Chapter 2, robust MTD is formulated as robust bilevel optimization to enhance the hiddenness while satisfying a pre-defined detection rate. To ensure feasibility and convergence, the nonlinear non-convex bilevel optimization is further relaxed into two successive semidefinite programmings using linear matrix inequalities and duality. Numerical experiments demonstrate that the shared information between the model and the data contributes to reducing the FPR of the data-driven detector, significantly enhancing its reliability. The additional operational cost incurred by the conservative nature of robust MTD becomes more economical due to the event-triggering mechanism of the data-driven detector and the informative data-driven attack uncertainty set.

Part II of the thesis studies the uncertainty and application of integrated E2E learning. In Chapter 4, the multi-step E2E learning problem is first formulated as a multi-level optimization problem. Compared to a single-level formulation, the optimal objective of multi-level optimization is equivalent to the cost of real-time operation. Then the uncertainty of E2E learning is classified into uncertainties on the input features and unpredictable parameters of down-stream optimizations. It is shown that neglecting either can cause generalization errors at the inference time. The main contribution of this chapter is to propose a unified robust optimization framework on anisotropic uncertainties. It is also proved that for piece-wise linear and quadratic down-stream optimizations, there exists an exact integer reformulation to find the exact worst input space uncertainty. The proposed

E2E-AT becomes a natural interconnection between adversarial training, differentiable optimization, and E2E learning.

Chapter 5 of the thesis applies the E2E learning framework to a “reverse” problem of unlearning the influence of data subset from the trained load forecasting model. Technically, this is the first time that machine unlearning is applied to the power system application. In a baseline algorithm, the influence of the unlearnt data is defined by the influence function and eliminated by second-order optimization. For the linear forecast model, the baseline algorithm is proved to be complete. We then show that complete unlearning can inevitably influence the statistic performance of the load forecaster. To balance completeness and performance, performance-aware unlearning is proposed to reweight the remaining dataset by its contribution to the performance of the test set. The implementation of E2E learning occurs when we modify the metric in PAMU as the generator cost. As a result, to balance completeness and task-aware generator cost, a tri-level optimization problem is formulated. The existence of the gradient of such task-aware objective has been proved theoretically and analytically. The simulation results verify that the proposed task-aware algorithm can significantly reduce the generator cost on the test dataset by compensating for the unlearning completeness.

6.2 Future Work

Due to the limited time, this thesis has only covered two main aspects of blending data and model for decision-making: the sequential and integrated formulations. Several theoretical and practical improvements could be explored in the future.

Firstly, the thesis has focused on down-stream tasks that can be modeled by convex optimizations within the E2E learning framework. Future research could address scenarios involving non-convex optimizations, such as those with integer variables or non-convex terms in real-time power system operations like UC or AC-OPF. Effectively and efficiently encoding non-convex optimizations as differentiable layers in the E2E framework remains an open research topic.

Secondly, the computational burden of training E2E models is extremely high compared to training with an accuracy-driven loss function, as the exact optimization problems need to be solved at each forward pass. The high computational burden restricts the use of E2E learning in large power grids. Future research could explore fast solution algorithms for specific optimization problems or find a trade-off between computation speed and solution optimality.

Third, this thesis discusses the E2E learning with deterministic forecasting and optimizations. As the motivation of E2E learning is to quantify the uncertainties of an unknown parameters in optimization, it is straightforward to extend it into probabilistic forecast followed by uncertain optimization and analyze its properties systematically. The stochastic setting can have a significant impact on the real-time application. For example, probabilistic load forecasting and two-stage stochastic power system operation can be used to deal with uncertainties of renewable generations and to quantify the risk of decision-making. Other viable options include data-driven uncertainty set forecast with robust optimization or ambiguity set forecast with distributionally robust optimization. The E2E learning framework could also be extended beyond supervised learning. Since the forecasting model adapts to different downstream tasks, it could be explored to integrate the dynamics of the power system into E2E learning, driven by model-based control or reinforcement learning.

For E2E-AT, a thorough investigation of the relationship between **multiple sources of uncertainty** could be an interesting future work, since conventional adversarial training usually has a single source of uncertainty. Theoretical analysis is needed and *Multi-task learning* [201], which trains simultaneously for various objectives, can be borrowed to handle multi-uncertainties. In addition, solving the exact attack vector is an MILP problem, which cannot be used for adversarial training due to its high complexity. Developing certified and tractable E2E-AT is also important for security purposes.

Finally, for the sequential data-driven and MTD detector for FDI attacks, there is potential in combining them into an integrated model. The transition from a regression problem to embedding a binary classification problem into an E2E learning framework also remains an open question for future exploration.

Appendices

Appendix A

Reproducibility

All data used for the thesis are publicly available. The results of the experiments are partially reproducible. Please refer to the following GitHub repositories for details:

1. Robust Moving Target Defence Against False Data Injection Attacks in Power Grids: `https://github.com/xuwkk/Robust_MTD`.
2. Blending Data and Physics Against False Data Injection Attack: An Event-Triggered Moving Target Defence Approach: `https://github.com/xuwkk/DDET-MTD`.
3. E2E-AT: A Unified Framework for Tackling Uncertainty in Task-aware End-to-end Learning: `https://github.com/xuwkk/E2E-AT`.
4. Task-Aware Machine Unlearning and Its Application in Load Forecasting: `https://github.com/xuwkk/task_aware_machine_unlearning`.

Appendix B

Mathematical Proofs

B.1 Proof to Proposition 1

The composite matrix of the original and perturbed Jacobian matrix (2.6) can be written as:

$$\begin{pmatrix} \boldsymbol{J} & \boldsymbol{J}' \end{pmatrix} = \boldsymbol{V} \cdot \begin{pmatrix} \boldsymbol{B} & -\boldsymbol{G} & \boldsymbol{B}' & -\boldsymbol{G} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{A}_r^{\cos} & \boldsymbol{0} \\ \boldsymbol{A}_r^{\sin} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_r^{\cos} \\ \boldsymbol{0} & \boldsymbol{A}_r^{\sin} \end{pmatrix}$$

Given the property of the matrix product, the rank of the composite matrix satisfies $\text{rank}((\boldsymbol{J} \quad \boldsymbol{J}')) \leq \min\{m, m, 2n\}$. If $m < 2n$, $\text{rank}((\boldsymbol{J} \quad \boldsymbol{J}')) \leq m < 2n$ no matter how the D-FACTS devices are altered. Therefore, the MTD cannot be complete if $m < 2n$.

B.2 Proof to Proposition 2

First, a β-MTD has $\|\boldsymbol{S}'_N \boldsymbol{a}_N\|_2 \geq \sqrt{\lambda_c(\beta)}$. The necessary condition then follows from $\|\boldsymbol{S}'_N \boldsymbol{a}_N\|_2 \leq \|\boldsymbol{S}_N\|_2 \|\boldsymbol{a}_N\|_2 = \|\boldsymbol{a}_N\|_2$.

Moreover, as $\boldsymbol{a}_N = \boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{a}$, it also gives $\|\boldsymbol{S}'_N\|_2 \|\boldsymbol{R}^{-\frac{1}{2}}\|_2 \|\boldsymbol{a}\|_2 = \|\boldsymbol{R}^{-\frac{1}{2}}\|_2 \|\boldsymbol{a}\|_2 \geq \sqrt{\lambda_c(\beta)}$. As $\|\boldsymbol{R}^{-\frac{1}{2}}\|_2 = \max \sigma(\boldsymbol{R}^{-\frac{1}{2}}) = \sigma_{min}^{-1}$, it can be derived that $\|\boldsymbol{a}\|_2 \geq \sigma_{min}\sqrt{\lambda_c(\beta)}$. Furthermore, if $\boldsymbol{R} = \text{diag}([\sigma, \sigma, \cdots, \sigma])$ is isotropic, it gives $\|\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{a}\|_2 = \sigma^{-1}\|\boldsymbol{a}\|_2 \geq \sqrt{\lambda_c(\beta)}$. Let

$\rho = \|\boldsymbol{a}\|_2/\sqrt{\sum_i^m \sigma_i^2}$. We can result in $\rho \geq \sqrt{\lambda_c(\beta)}/\sqrt{m}$.

B.3 Proof to Proposition 3

According to Definition 1, the weakest point $\boldsymbol{j}_N^* \in \mathcal{J}_N, \|\boldsymbol{j}_N^*\|_2 = 1$ can be derived by

$$\begin{aligned} \boldsymbol{j}_N^* = & \arg\min_{\substack{\boldsymbol{j}_N \in \mathcal{J}_N \\ \|\boldsymbol{j}_N\|_2 = 1}} \sqrt{\lambda_{eff}} \\ = & \arg\min_{\substack{\boldsymbol{j}_N \in \mathcal{J}_N \\ \|\boldsymbol{j}_N\|_2 = 1}} \frac{\|\boldsymbol{j}_N - \boldsymbol{P}_N' \boldsymbol{j}_N\|_2}{\|\boldsymbol{j}_N\|_2} \\ = & \arg\min_{\substack{\boldsymbol{j}_N \in \mathcal{J}_N \\ \|\boldsymbol{j}_N\|_2 = 1}} \sin\angle\{\boldsymbol{j}_N, \boldsymbol{P}_N' \boldsymbol{j}_N\} \end{aligned} \tag{B.1}$$

Note that the triangle relationship within the sides $\|\boldsymbol{j}_N\|$, $\|\boldsymbol{P}_N'\boldsymbol{j}_N\|$, and $\|\boldsymbol{j}_N - \boldsymbol{P}_N'\boldsymbol{j}_N\|$ and the ratio in (B.1) is the sine of the angle between the vectors $\boldsymbol{j}_N$ and $\boldsymbol{P}_N'\boldsymbol{j}_N$. Basing on the definition of principal angle (2.11), the sine of the angle is minimized when $\angle\{\boldsymbol{j}_N, \boldsymbol{P}_N'\boldsymbol{j}_N\} = \theta_1$. The minimum principal angle is achieved when $\boldsymbol{j}_N$ and $\boldsymbol{P}_N'\boldsymbol{j}_N$ are reciprocal such that $\boldsymbol{j}_N = \boldsymbol{u}_1$ and $\boldsymbol{P}_N'\boldsymbol{j}_N = \boldsymbol{P}_N'\boldsymbol{u}_1 = \cos\theta_1 \boldsymbol{v}_1$ [136, 202].

Moreover, the worst-case detection rate is achieved when attacking on $\boldsymbol{u}_1$ such that

$$\lambda_{min} = \|a\boldsymbol{u}_1 - a\cos\theta_1\boldsymbol{v}_1\|_2^2 = a^2\sin^2\theta_1$$

B.4 Proof to Lemma 1

Rewrite the non-centrality parameter as

$$\begin{aligned} \sqrt{\lambda} = & \|(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^T)\boldsymbol{U}\boldsymbol{c}\|_2 \\ = & \|(\boldsymbol{U} - \boldsymbol{V}\Gamma)\boldsymbol{c}\|_2 \\ = & \|\left((\boldsymbol{U}_1, \boldsymbol{U}_{23}) - (\boldsymbol{V}_1\Gamma_1, \boldsymbol{V}_{23}\Gamma_{23})\right)\boldsymbol{c}\|_2 \end{aligned} \tag{B.2}$$

As $\boldsymbol{U}_1 = \boldsymbol{V}_1$ and $\Gamma_1 = \boldsymbol{I}$, (B.2) can be reduced to $\sqrt{\lambda} = (\boldsymbol{U}_{23} - \boldsymbol{V}_{23}\Gamma_{23})\boldsymbol{c}_{23}$ which does not depend on $\boldsymbol{c}_1$.

B.5 Proof to Proposition 4

To start, the Schur complement [203] is given as follows.

Theorem 1. *Given any symmetric matrix* $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \star & \boldsymbol{C} \end{bmatrix}$, *if* $\boldsymbol{C}$ *is invertible, the following two conditions are equivalent: (1) If* $\boldsymbol{C} \succ 0$, *then* $\boldsymbol{Z} \succeq 0$; *(2)* $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{C}^{-1}\boldsymbol{B}^T \succeq 0$.

Proposition 11. *Given any symmetric matrix* $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \star & \boldsymbol{C} \end{bmatrix}$, *the following two conditions are equivalent: (1)* $\boldsymbol{Z} \succeq 0$; *(2)* $\boldsymbol{C} \succeq 0$, $(\boldsymbol{I} - \boldsymbol{C}\boldsymbol{C}^{\dagger})\boldsymbol{B}^T = 0$, $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{C}^{\dagger}\boldsymbol{B}^T \succeq 0$.

First, the inner problem of (3.18) can be written as:

$$\begin{aligned} \min \quad & \|\boldsymbol{S}_0'\boldsymbol{H}_1\boldsymbol{c}'\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{c}' - \bar{\boldsymbol{c}}\|_2^2 \le \varrho^2 \end{aligned} \tag{B.3}$$

where $\boldsymbol{S}_0' = \mathcal{S}(\boldsymbol{H}_0')$ and the Lagrangian of (B.3) is written as:

$$\mathcal{L}(\boldsymbol{c}', \nu) = \boldsymbol{c}'^T \left(\boldsymbol{H}_1^T\boldsymbol{S}_0'\boldsymbol{H}_1 + \nu\boldsymbol{I}\right)\boldsymbol{c}' + (-2\nu\bar{\boldsymbol{c}}^T)\boldsymbol{c}' + \nu(\bar{\boldsymbol{c}}^T\bar{\boldsymbol{c}} - \varrho^2) \tag{B.4}$$

Based on (B.4) and denoting $\boldsymbol{M} = \boldsymbol{H}_1^T\boldsymbol{S}_0'\boldsymbol{H}_1 + \nu\boldsymbol{I}$, the dual function of (B.3) can be analytically written as

$$\begin{aligned} g(\nu) &= \inf_{\boldsymbol{c}'} \mathcal{L}(\boldsymbol{c}', \nu) \\ &= \begin{cases} -(\nu\bar{\boldsymbol{c}})^T\boldsymbol{M}^{\dagger}(\nu\bar{\boldsymbol{c}}) + \nu(\bar{\boldsymbol{c}}^T\bar{\boldsymbol{c}} - \varrho^2) & \boldsymbol{M} \succeq 0, \nu\bar{\boldsymbol{c}} \in \mathrm{Col}(\boldsymbol{M}) \\ -\infty & \text{otherwise} \end{cases} \end{aligned} \tag{B.5}$$

Let $-(\nu\bar{\boldsymbol{c}})^T\boldsymbol{M}^\dagger(\nu\bar{\boldsymbol{c}}) + \nu(\bar{\boldsymbol{c}}^T\bar{\boldsymbol{c}} - \varrho^2) \geq \omega$. The dual problem of (B.3) becomes:

$$\begin{aligned} \max_{\nu,\omega} \quad & \omega \\ \text{subject to} \quad & \nu \geq 0 \\ & \nu(\bar{\boldsymbol{c}}^T\bar{\boldsymbol{c}} - \varrho^2) - \omega - (\nu\bar{\boldsymbol{c}})^T\boldsymbol{M}^\dagger(\nu\bar{\boldsymbol{c}}) \geq 0 \\ & \boldsymbol{M} \succeq 0 \\ & \nu\bar{\boldsymbol{c}} \in \mathrm{Col}(\boldsymbol{M}) \end{aligned} \tag{B.6}$$

Note that the last constraint of (B.6) can be rewritten as $\boldsymbol{M}\boldsymbol{M}^\perp\nu\bar{\boldsymbol{c}} = \nu\bar{\boldsymbol{c}}$. Applying Proposition 11, the dual problem can be rewritten as

$$\begin{aligned} \max_{\nu,\omega} \quad & \omega \\ \text{subject to} \quad & \nu \geq 0 \\ & \begin{bmatrix} \nu(\bar{\boldsymbol{c}}^T\bar{\boldsymbol{c}} - \varrho^2) - \omega & (\nu\bar{\boldsymbol{c}})^T \\ \star & \boldsymbol{M} \end{bmatrix} \succeq 0 \end{aligned} \tag{B.7}$$

The strong duality between (B.3) and (B.7) holds as long as $\mathcal{C} \neq \emptyset$ [157]. Consequently, the robust stage one problem (3.18) becomes:

$$\max_{\boldsymbol{b}',\mu,\omega} \quad \omega \tag{B.8a}$$

$$\text{subject to} \quad [\boldsymbol{b}'] - [\boldsymbol{b}^-] \succeq 0, [\boldsymbol{b}^+] - [\boldsymbol{b}'] \succeq 0 \tag{B.8b}$$

$$\nu \geq 0 \tag{B.8c}$$

$$\begin{bmatrix} \nu(\bar{\boldsymbol{c}}^T\bar{\boldsymbol{c}} - \varrho^2) - \omega & (\nu\bar{\boldsymbol{c}})^T \\ \star & \boldsymbol{M} \end{bmatrix} \succeq 0 \tag{B.8d}$$

Note that $\boldsymbol{M} = \nu\boldsymbol{I} + \boldsymbol{H}_1^T\boldsymbol{H}_1 - \boldsymbol{H}_1^T\boldsymbol{H}_0'(\boldsymbol{H}_0'^T\boldsymbol{H}_0')^{-1}\boldsymbol{H}_0'^T\boldsymbol{H}_1$ is nonlinear in the decision variable $\boldsymbol{b}'$. In Theorem 1, define $\boldsymbol{A} := \begin{bmatrix} \nu(\bar{\boldsymbol{c}}^T\bar{\boldsymbol{c}} - \varrho^2) - \omega & (\nu\bar{\boldsymbol{c}})^T \\ \star & \nu\boldsymbol{I} + \boldsymbol{H}_1^T\boldsymbol{H}_1 \end{bmatrix}$, $\boldsymbol{B} :=$

$\begin{bmatrix} \mathbf{0} \\ \boldsymbol{H}_1^T \boldsymbol{H}_0' \end{bmatrix}$, and $\boldsymbol{C} := \boldsymbol{H}_0'^T \boldsymbol{H}_0'$ in (B.8d). Since $\boldsymbol{C} > 0$ and Theorem 1, the constraint (B.8d) becomes (3.19d), which finalizes the proof.

B.6 Proof to Proposition 5

First, the following sufficient condition holds for any matrices $\boldsymbol{E}, \boldsymbol{E}_0$ with the same dimension [204]:

$$\boldsymbol{E}_0^T \boldsymbol{E} + \boldsymbol{E}^T \boldsymbol{E}_0 - \boldsymbol{E}_0^T \boldsymbol{E}_0 \succeq 0 \Rightarrow \boldsymbol{E}^T \boldsymbol{E} \succeq 0$$

Define $\boldsymbol{E} = \boldsymbol{C}^N + \boldsymbol{V}^N[\boldsymbol{b}']\boldsymbol{A}_r^c$ and $\boldsymbol{E}_0 = \boldsymbol{V}^N[\boldsymbol{b}_k]\boldsymbol{A}_r^c$. Replacing $\boldsymbol{H}_0'^T \boldsymbol{H}_0'$ in (3.19d) by $\boldsymbol{H}_{\text{update}} = \boldsymbol{E}_0^T \boldsymbol{E} + \boldsymbol{E}^T \boldsymbol{E}_0 - \boldsymbol{E}_0^T \boldsymbol{E}_0$ finalizes the proof.

B.7 Proof to Proposition 6

The dual function (B.5) is the lower bound of the primary function, e.g. $g(\nu) \leq \|\boldsymbol{S}_0' \boldsymbol{H}_1 \boldsymbol{c}'\|_2^2$ for $\forall \boldsymbol{c}' \in \mathcal{C}$. Therefore, a sufficient condition for (3.16c) is $g(\nu) \geq \omega$. Note that $\lambda_c(\rho)$ is replaced by constant ω in stage two. Therefore, Proposition 6 can be proved similarly to Proposition 4. Furthermore, define the cost of (3.16) as $(\boldsymbol{b}' - \boldsymbol{b}_0)^T \boldsymbol{H}_{\text{hid}}^T \boldsymbol{I} \boldsymbol{H}_{\text{hid}} (\boldsymbol{b}' - \boldsymbol{b}_0) \leq \phi$. Applying Theorem 1 on $\boldsymbol{I}$ gives (3.22c).

B.8 Proof to Proposition 7

First, $M(\hat{\boldsymbol{z}}_1^\star) \leq M(\hat{\boldsymbol{z}}_2^\star)$ can be directly verified since (4.9) has a tighter constraint on the optimality condition in the lower level problem than it in (4.8). For fixed $\boldsymbol{\theta}_2^\star$, the optimal decision $\hat{\boldsymbol{z}}_r^\star(\boldsymbol{\theta}_2^\star)$ is achieved when each subproblem, represented as the lower level problem in (4.9), achieves its optimum. Therefore, $\ell(\hat{\boldsymbol{z}}_2^{i\star}; \boldsymbol{y}^i) = \ell(\hat{\boldsymbol{z}}_r^{i\star}(\boldsymbol{\theta}_2^\star); \boldsymbol{y}^i)$ and $M(\hat{\boldsymbol{z}}_2^\star) = M(\hat{\boldsymbol{z}}_r^\star(\boldsymbol{\theta}_2^\star))$. It also shows that $(\boldsymbol{\theta}_2^\star, \boldsymbol{z}_r^\star(\boldsymbol{\theta}_2^\star))$ is the minimizer of (4.9). Note that multiple global minimizers are also satisfied. Since $(\boldsymbol{\theta}_1^\star, \boldsymbol{z}_r^\star(\boldsymbol{\theta}_1^\star))$ is also feasible with (4.9), it gives $M(\hat{\boldsymbol{z}}_r^\star(\boldsymbol{\theta}_2^\star)) \leq M(\hat{\boldsymbol{z}}_r^\star(\boldsymbol{\theta}_1^\star))$, which finalizes the proof.

B.9 Proof to Proposition 8

We give the proof by first introducing the complementary linearization [205,206]:

Proposition 12. *The complementary condition $a \geq 0, b \geq 0, a \cdot b = 0$ can be replaced by*

$$a \geq 0,\ b \geq 0,\ a \leq \psi M,\ b \leq (1-\psi)M,\ \psi \in \{0,1\}$$

where M is a large enough constant.

For the optimal primal-dual pair $(\boldsymbol{z}_{i+1}, \boldsymbol{\lambda}_{i+1}, \boldsymbol{\nu}_{i+1})$, the Karush–Kuhn–Tucker (KKT) condition [157] gives that

$$\boldsymbol{Q}\boldsymbol{z}_{i+1} + \boldsymbol{q} + \boldsymbol{A}^T\boldsymbol{\lambda}_{i+1} + \boldsymbol{C}^T\boldsymbol{\nu}_{i+1} = 0 \tag{B.9a}$$

$$\boldsymbol{C}\boldsymbol{z}_{i+1} + \boldsymbol{H}\boldsymbol{z}_i - \boldsymbol{d} = 0 \tag{B.9b}$$

$$\operatorname{diag}(\boldsymbol{\lambda}_{i+1})(\boldsymbol{A}\boldsymbol{z}_{i+1} + \boldsymbol{G}\boldsymbol{z}_i + \boldsymbol{b}) = 0 \tag{B.9c}$$

$$\boldsymbol{A}\boldsymbol{z}_{i+1} + \boldsymbol{G}\boldsymbol{z}_i - \boldsymbol{b} \leq 0 \tag{B.9d}$$

$$\boldsymbol{\lambda}_{i+1} \geq 0 \tag{B.9e}$$

Each of the lower-level problems can be equivalently written in this form. Since the complementary condition states that at least one of the inequality constraints or the dual variable equals zero, (B.9c), (B.9d), and (B.9e) can be rewritten by Proposition 12, which finalizes the proof.

B.10 Proof to Proposition 9

We prove Proposition 9 by proving a more general Proposition 13. To start, consider the following QP:

$$\begin{aligned} \boldsymbol{x}^\star = &\arg\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{q}^T\boldsymbol{x} \\ &\text{s.t. } \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} + \boldsymbol{g}(\boldsymbol{z}) \leq \boldsymbol{0} \\ &\boldsymbol{C}\boldsymbol{x} + \boldsymbol{d} + \boldsymbol{h}(\boldsymbol{z}) = \boldsymbol{0} \end{aligned} \tag{B.10}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$, $\boldsymbol{q} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{C} \in \mathbb{R}^{p\times n}$, $\boldsymbol{d} \in \mathbb{R}^p$, and $\boldsymbol{z} \in \mathbb{R}^q$. $\boldsymbol{g} : \mathbb{R}^q \to \mathbb{R}^m$ and $\boldsymbol{h} : \mathbb{R}^q \to \mathbb{R}^p$ are functions on $\boldsymbol{z}$, representing the perturbation parameters. Apart from the linear parametric inequality constraints in (5.17), we also include the linear parametric term $\boldsymbol{g}(\boldsymbol{z})$ in the inequality constraint for generalization purposes (and it also gives the same conclusion). We call (B.10) affine-parametric, since the parametric terms $\boldsymbol{g}(\boldsymbol{z})$ and $\boldsymbol{h}(\boldsymbol{z})$ are affine in the inequality and equality constraints.

Proposition 13. *Given an affine parametric QP* (B.10)*, the optimal primal and dual pair* $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$ *is an affine function of the parameter* $(\boldsymbol{g}(\boldsymbol{z}), \boldsymbol{h}(\boldsymbol{z}))$ *if 1).* $\boldsymbol{Q}$ *is positive definite; and 2). the linear independent constraint qualification (LICQ) is satisfied at* $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$*.*

Proof. First, the LICQ states that the gradient of the active constraints (including all equality constraints and active inequality constraints) are linearly independent [207]. Therefore, $\boldsymbol{C}$ is full row rank.Second, the equality Karush–Kuhn–Tucker (KKT) conditions [207] can be denoted as:

$$\boldsymbol{G}(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star, \boldsymbol{z}) = \begin{pmatrix} \boldsymbol{Q}\boldsymbol{x}^\star + \boldsymbol{q} + \boldsymbol{A}^T\boldsymbol{\lambda}^\star + \boldsymbol{C}^T\boldsymbol{\nu}^\star \\ \text{diag}(\boldsymbol{\lambda}^\star)(\boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{b} + \boldsymbol{g}(\boldsymbol{z})) \\ \boldsymbol{C}\boldsymbol{x}^\star + \boldsymbol{d} + \boldsymbol{h}(\boldsymbol{z}) \end{pmatrix} = \boldsymbol{0}$$

We divide the proofs by the existence of active constraints.

When there are no active inequality constraints, $\boldsymbol{\lambda}^\star = \boldsymbol{0}$ due to complementary slackness. Since $\boldsymbol{Q}$ is positive definite, the stationary condition gives $\boldsymbol{x}^\star = -\boldsymbol{Q}^{-1}(\boldsymbol{q} + \boldsymbol{C}^T\boldsymbol{\nu}^\star)$. From the equality constraint, it can be derived that $\boldsymbol{\nu}^\star = (\boldsymbol{C}\boldsymbol{Q}^{-1}\boldsymbol{C}^T)^{-1}(-\boldsymbol{C}\boldsymbol{Q}^{-1}\boldsymbol{q} + \boldsymbol{d} +$

$\boldsymbol{h}(\boldsymbol{z}))$. Note that $\boldsymbol{C}\boldsymbol{Q}^{-1}\boldsymbol{C}^T$ is positive definite (thus invertible). Let $\hat{\boldsymbol{C}} = \boldsymbol{Q}^{-1}\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{Q}^{-1}\boldsymbol{C}^T)^{-1}$, the analytical form for $\boldsymbol{x}^\star$ can be written as

$$\boldsymbol{x}^\star = (-\boldsymbol{Q}^{-1} + \hat{\boldsymbol{C}}\boldsymbol{C}\boldsymbol{Q}^{-1})\boldsymbol{q} - \hat{\boldsymbol{C}}(\boldsymbol{d} + \boldsymbol{h}(\boldsymbol{z})) \tag{B.11}$$

which is affine in $\boldsymbol{h}(\boldsymbol{z})$.

When there exist some active inequality constraints, let $\tilde{\boldsymbol{\lambda}}$, $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{b}}$, and $\tilde{\boldsymbol{g}}(\boldsymbol{z})$ be the sub-matrices whose rows are indexed by the active constraints. Therefore, $\boldsymbol{A}^T\boldsymbol{\lambda}^\star = \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\lambda}}^\star$ and the active inequality constraint becomes:

$$\tilde{\boldsymbol{A}}\boldsymbol{x}^\star + \tilde{\boldsymbol{b}} + \tilde{\boldsymbol{g}}(\boldsymbol{z}) = 0 \tag{B.12}$$

Since $\boldsymbol{Q}$ is positive definite, the stationary condition gives that

$$\boldsymbol{x}^\star = -\boldsymbol{Q}^{-1}(\boldsymbol{q} + \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{\lambda}}^\star + \boldsymbol{C}^T\boldsymbol{\nu}^\star) \tag{B.13}$$

Plugging (B.13) into (B.12) and the equality condition gives the following matrix form:

$$\boldsymbol{\mathcal{Q}}\begin{pmatrix}\tilde{\boldsymbol{\lambda}}^\star \\ \boldsymbol{\nu}^\star\end{pmatrix} = \underbrace{\begin{pmatrix}-\tilde{\boldsymbol{A}}\boldsymbol{Q}^{-1}\boldsymbol{q} + \tilde{\boldsymbol{b}} + \tilde{\boldsymbol{g}}(\boldsymbol{z}) \\ -\boldsymbol{C}\boldsymbol{Q}^{-1}\boldsymbol{q} + \boldsymbol{d} + \boldsymbol{h}(\boldsymbol{z})\end{pmatrix}}_{\boldsymbol{r}(\boldsymbol{z})} \tag{B.14}$$

where

$$\begin{aligned}\boldsymbol{\mathcal{Q}} &= \begin{pmatrix}\tilde{\boldsymbol{A}}\boldsymbol{Q}^{-1}\tilde{\boldsymbol{A}}^T & \tilde{\boldsymbol{A}}\boldsymbol{Q}^{-1}\boldsymbol{C}^T \\ \boldsymbol{C}\boldsymbol{Q}^{-1}\tilde{\boldsymbol{A}}^T & \boldsymbol{C}\boldsymbol{Q}^{-1}\tilde{\boldsymbol{C}}^T\end{pmatrix} \\ &= \begin{pmatrix}\tilde{\boldsymbol{A}} \\ \boldsymbol{C}\end{pmatrix}\boldsymbol{Q}^{-1}\begin{pmatrix}\tilde{\boldsymbol{A}}^T & \boldsymbol{C}^T\end{pmatrix}\end{aligned}$$

Due to LICQ, $(\tilde{\boldsymbol{A}}^T, \boldsymbol{C}^T)$ is full column rank. Therefore, $\boldsymbol{\mathcal{Q}}$ is positive definite and from

(B.14)

$$\begin{pmatrix} \tilde{\boldsymbol{\lambda}}^\star \\ \boldsymbol{\nu}^\star \end{pmatrix} = \boldsymbol{\mathcal{Q}}^{-1}\boldsymbol{r}(\boldsymbol{z}) \tag{B.15}$$

which is affine in $(\tilde{\boldsymbol{g}}(\boldsymbol{z})^T, \boldsymbol{h}(\boldsymbol{z})^T)^T$. Consequently, plugging (B.15) into (B.13) gives

$$\boldsymbol{x}^\star = -\boldsymbol{Q}^{-1}\left(\boldsymbol{q} + (\tilde{\boldsymbol{A}}^T, \boldsymbol{C}^T)\boldsymbol{\mathcal{Q}}^{-1}\boldsymbol{r}(\boldsymbol{z})\right) \tag{B.16}$$

which is affine in $(\tilde{\boldsymbol{g}}(\boldsymbol{z})^T, \boldsymbol{h}(\boldsymbol{z})^T)^T$.

□

Since the optimal solution of every QP satisfying Proposition 13 is an affine function of the parameter ((B.11) and (B.16)), the gradients of the convex layers in the dispatch and re-dispatch problems exist and can be analytically written regardless of the perturbed parameter.

B.11 Proof to Proposition 10

Let $\boldsymbol{f}(\cdot;\boldsymbol{\theta}^\star_{\text{FE}})$ be the trained feature extractor on the pre-train dataset. Let $\boldsymbol{X}_{\text{sen}} \in \mathbb{R}^{N_{\text{sen}}\times d}$ be the extracted feature of $\mathcal{D}_{\text{sen}}$ as input to the Linear Layer 2. N_{sen} is the number of user sensitive data and d is the output size of feature extractor. Note that $d \ll N_{\text{sen}}$ and $\boldsymbol{X}_{\text{sen}}$ is full column rank by the condition. Meanwhile, let $\boldsymbol{Y}_{\text{sen}} \in \mathbb{R}^{N_{\text{sen}}\times n}$ be the ground truth load over n participants. The parameter of Linear Layer 2 is denoted as $\boldsymbol{\Theta} \in \mathbb{R}^{d\times n}$.

Let $\boldsymbol{y}_{\cdot,i} \in \mathbb{R}^{N_{\text{sen}}}$ and $\boldsymbol{\theta}_{\cdot,i} \in \mathbb{R}^d$ be the i-th column of $\boldsymbol{Y}_{\text{sen}}$ and $\boldsymbol{\Theta}$, respectively. The fine-tuning objective can be written as

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N_{\text{sen}}\cdot n}\sum_{i=1}^{n} \|\boldsymbol{y}_{\cdot,i} - \boldsymbol{X}_{\text{sen}}\boldsymbol{\theta}_i\|_2^2 \tag{B.17}$$

Now define $\hat{\boldsymbol{X}}_{\text{sen}} = \text{diag}([\underbrace{\boldsymbol{X}_{\text{sen}}, \cdots, \boldsymbol{X}_{\text{sen}}}_{n}]) \in \mathbb{R}^{N_{\text{sen}}n\times dn}$ as a block diagonal matrix packed by n $\boldsymbol{X}_{\text{sen}}$s. $\hat{\boldsymbol{Y}}_{\text{sen}} = [\boldsymbol{y}^T_{\cdot,1}, \cdots, \boldsymbol{y}^T_{\cdot,n}]^T \in \mathbb{R}^{N_{\text{sen}}n}$ and $\hat{\boldsymbol{\Theta}} = [\boldsymbol{\theta}^T_{\cdot,1}, \cdots, \boldsymbol{\theta}^T_{\cdot,n}]^T \in \mathbb{R}^{dn}$ be the flattened version of $\boldsymbol{Y}_{\text{sen}}$ and $\boldsymbol{\Theta}$, respectively. It can be verified that (B.17) is equivalent to

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N_{\text{sen}} \cdot n} \left(\hat{\boldsymbol{\Theta}}^T \hat{\boldsymbol{X}}_{\text{sen}}^T \hat{\boldsymbol{X}}_{\text{sen}} \hat{\boldsymbol{\Theta}} - 2 \boldsymbol{Y}_{\text{sen}}^T \hat{\boldsymbol{X}}_{\text{sen}} \hat{\boldsymbol{\Theta}} + \hat{\boldsymbol{Y}}_{\text{sen}}^T \hat{\boldsymbol{Y}}_{\text{sen}} \right) \tag{B.18}$$

Since $\boldsymbol{X}_{\text{sen}}$ is full column rank, $\hat{\boldsymbol{X}}_{\text{sen}}^T \hat{\boldsymbol{X}}_{\text{sen}}$ is positive definite. Therefore, (B.18) and (B.17) are quadratic with unique global minimizer.

Appendix C

Supplementary Material for Robust Moving Target Defence Against False Data Injection Attacks in Power Grids

C.1 Normalised Measurement Vectors and Matrices

We consider measurement noise follows independent Gaussian distribution which is not necessarily isotropic. Let $\boldsymbol{z}_N = \boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{z}$, $\boldsymbol{e}_N = \boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{e}$, and $\boldsymbol{J}_N = \boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{J}$. The measurement equation becomes $\boldsymbol{z}_N = \boldsymbol{J}_N\boldsymbol{\theta} + \boldsymbol{e}_N$. $\boldsymbol{P}_{\boldsymbol{J}}$, which is defined on $\langle\, , \rangle_{\boldsymbol{R}^{-\frac{1}{2}}}$, now becomes $\boldsymbol{P}_{\boldsymbol{J}_N} = \boldsymbol{J}_N(\boldsymbol{J}_N^T\boldsymbol{J}_N)^{-1}\boldsymbol{J}_N^T$, defined on $\langle\, , \rangle$. Similarly, $\boldsymbol{S}_{\boldsymbol{J}_N} = \boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{J}_N}$. It is easy to show that $\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{S}_{\boldsymbol{J}} = \boldsymbol{S}_{\boldsymbol{J}_N}\boldsymbol{R}^{-\frac{1}{2}}$. As a result, $\boldsymbol{r}(\boldsymbol{z}_N) = \boldsymbol{S}_{\boldsymbol{J}_N}\boldsymbol{e}_N$ follows (approximately) standard normal distribution $\boldsymbol{r}(\boldsymbol{z}_N) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. For convenience, we write $\boldsymbol{P}_{\boldsymbol{J}_N}$ and $\boldsymbol{S}_{\boldsymbol{J}_N}$ as $\boldsymbol{P}_N$ and $\boldsymbol{S}_N$ in short of the main content.

C.2 D-FACTS Devices Placement

A modified minimum edge covering algorithm is proposed to find the smallest number of D-FACTS devices covering all buses while satisfying the minimum k condition. The pseudocode is given by Algorithm 4. In detail, the inputs to the proposed MTD deployment algorithm are the grid information $\mathcal{G}(\mathcal{N},\mathcal{E})$ and the output is branch set $\mathcal{E}_D$. On lines 1-2, CB represents the function to calculate the set of cycle bases of a given graph. The algorithm 4 then removes any buses that are not included by cycle basis (thus not in any loops) and the corresponding branches from the grid $\mathcal{G}$. In line 3-4, the minimum edge covering (MEC) problem is solved. Given the power grid topology, MEC firstly runs the maximum (cardinality) matching algorithm to find the maximum branch set whose ending buses are not incident to each other [138]. The maximum matching is found by Edmonds' BLOSSOM algorithm where the size of the initial empty matching is increased iteratively along the so-called augmenting path spotted by blossom contraction [138]. After constructing the maximum matching, a greedy algorithm is performed to add any uncovered buses to the maximum matching set. The resulting set of branches becomes $\mathcal{E}_D$, the minimum edge covering set where each bus is connected to at least one branch. Lines 5-15 guarantee the minimum k requirement where it breaks the edge in any identified cycle bases in $\overline{\mathcal{G}}_2$. At last, line 11-13 is added to avoid adding any new loop in $\overline{\mathcal{G}}_1$.

Algorithm 4: D-FACTS Devices Placement Algorithm

```
Input  : grid topology $\mathcal{G}(\mathcal{N}, \mathcal{E})$
Output: branch set with D-FACTS devices $\mathcal{E}_D$
$\mathcal{L} = \text{CB}(\mathcal{G})$; /* find the circle basis */
Find buses $\mathcal{N}_1$ not in $\mathcal{L}$. Remove $\mathcal{N}_1$ and the incident branches from $\mathcal{G}$. Name the
  resulting graph as $\overline{\mathcal{G}}(\overline{\mathcal{N}}, \overline{\mathcal{E}})$;
$\mathcal{E}_{min} = \text{MEC}(\overline{\mathcal{G}})$, construct $\overline{\mathcal{G}}_1(\overline{\mathcal{N}}, \mathcal{E}_{min})$ and $\overline{\mathcal{G}}_2(\overline{\mathcal{N}}, \mathcal{E}_r)$ with $\mathcal{E}_r = \overline{\mathcal{E}} \setminus \mathcal{E}_{min}$;
$\mathcal{L}_2 = \text{CB}(\overline{\mathcal{G}}_2)$ /* loops in non D-FACTs graph */
for loop in $\mathcal{L}_2$ do
    for $e$ in loop do
        Construct $\overline{\mathcal{G}}_1(\overline{\mathcal{N}}, \mathcal{E}_{min})$ and $\overline{\mathcal{G}}_2(\overline{\mathcal{N}}, \mathcal{E}_r)$ where $\mathcal{E}_{min} \leftarrow \mathcal{E}_{min} + e$ and
          $\mathcal{E}_r \leftarrow \mathcal{E}_r - e$;
        $\mathcal{L}_1 = \text{CB}(\overline{\mathcal{G}}_1)$;
        /* loops in D-FACTs graph */
        if $\mathcal{L}_1 = \varnothing$ then
            break
        else
            $\overline{\mathcal{G}}_1(\overline{\mathcal{N}}, \mathcal{E}_{min})$ and $\overline{\mathcal{G}}_2(\overline{\mathcal{N}}, \mathcal{E}_r)$ where $\mathcal{E}_{min} \leftarrow \mathcal{E}_{min} - e$ and $\mathcal{E}_r \leftarrow \mathcal{E}_r + e$;
        end
    end
end
```

Appendix D

Supplementary Material for Blending Data and Physics Against False Data Injection Attack: An Event-Triggered Moving Target Defence Approach

D.1 Convert the Range of Reactance into Susceptance

Let the branch i have resistance $\boldsymbol{r}_i \in (0, \infty)$ and reactance $\boldsymbol{x}_i \in (0, \infty)$. The susceptance $\boldsymbol{b}_i$ can be determined as:

$$\boldsymbol{b}_i(\boldsymbol{x}_i) = \frac{-\boldsymbol{x}_i}{\boldsymbol{r}_i^2 + \boldsymbol{x}_i^2}$$

Therefore, $\boldsymbol{b}_i$ decreases monotonically on $(0, \boldsymbol{r}_i)$ and increases on $(\boldsymbol{r}_i, \infty)$. Considering the permissible range of $\boldsymbol{x}_i \in [\boldsymbol{x}_i^-, \boldsymbol{x}_i^+]$, the permissible range of $\boldsymbol{b}_i$ can be determined. For $\boldsymbol{r}_i \in [\boldsymbol{x}_i^-, \boldsymbol{x}_i^+]$, $\boldsymbol{b}^- = \frac{-\boldsymbol{r}_i}{\boldsymbol{r}_i^2+\boldsymbol{x}_i^2}$, $\boldsymbol{b}^+ = \max\left(\frac{-\boldsymbol{x}_i^+}{\boldsymbol{r}_i^2+{\boldsymbol{x}_i^+}^2}, \frac{-\boldsymbol{x}_i^-}{\boldsymbol{r}_i^2+{\boldsymbol{x}_i^-}^2}\right)$; for $\boldsymbol{r}_i \notin [\boldsymbol{x}_i^-, \boldsymbol{x}_i^+]$, $\boldsymbol{b}^- = \min\left(\frac{-\boldsymbol{x}_i^+}{\boldsymbol{r}_i^2+{\boldsymbol{x}_i^+}^2}, \frac{-\boldsymbol{x}_i^-}{\boldsymbol{r}_i^2+{\boldsymbol{x}_i^-}^2}\right)$, $\boldsymbol{b}^+ = \max\left(\frac{-\boldsymbol{x}_i^+}{\boldsymbol{r}_i^2+{\boldsymbol{x}_i^+}^2}, \frac{-\boldsymbol{x}_i^-}{\boldsymbol{r}_i^2+{\boldsymbol{x}_i^-}^2}\right)$.

D.2 Hidden and Effective MTD Algorithm

The hidden and effective MTD algorithm is summarized in Algorithm 5 in detail. The inputs of the algorithm $\mathcal{B}$, $\mathcal{C}$, $\boldsymbol{C}^N$, $\boldsymbol{V}^N$, $\boldsymbol{A}_r^c$, $\boldsymbol{H}_1$, $\boldsymbol{H}_{\text{hid}}$, and λ_c' have been defined in the main content. Furthermore, define tol_{one} and tol_{two} as the tolerance of stage-one and stage-two problem, respectively. Define ite_{one} and ite_{two} as the maximum iteration step in stage-one and stage-two problem, respectively. Meanwhile, let no be the multi-run number. The output of this algorithm is the set-point of the D-FATCS devices, denoted as $\boldsymbol{b}_{\text{mtd}}$.

As shown by Algorithm 5 lines 1-15, the optimal solution of (3.21) can replace $\boldsymbol{b}_{k+1}$ and the iteration continues. The iteration ends if the improvement in ω is limited or the step no is achieved. The multi-run solutions are stored to be further used in stage two. As in Algorithm 5 lines 17-25, if the maximum state one objective ω^* is smaller than threshold λ_c', the stage-two problem starts at the corresponding susceptance and reduces the threshold into ω^*. If the threshold is achieved by stage one, a candidate feasible susceptance set $\mathcal{D}^{\text{two}}$ is established, and multi-runs are implemented on this candidate set. As shown in lines 27-37, for each candidate $\boldsymbol{b}$, (3.23) is solved iteratively until convergence. Finally, the susceptance $\boldsymbol{b}_{\text{mtd}}$ with the smallest cost is returned in line 38.

Algorithm 5: Hidden and Effective MTD Algorithm

```
Input  : $\mathcal{B}$, $\mathcal{C}$, $\boldsymbol{C}^N$, $\boldsymbol{V}^N$, $\boldsymbol{A}_r^c$, $\boldsymbol{H}_1$, $\boldsymbol{H}_{\text{hid}}$, $\lambda_c'$, $tol_{one}$, $tol_{two}$, $ite_{one}$, $ite_{two}$, $no$
Output: $\boldsymbol{b}_{\text{mtd}}$
/* Stage-One Algorithm                                                      */
$\mathcal{D}^{\text{one}} = \{\cdot\}$, $\mathcal{G}^{\text{one}} = \{\cdot\}$ /* Store multi-run results               */
$i = 0$
while $i \leq no$ do
    $k = 0$, $\omega_k = 0$
    Random generate $\boldsymbol{b}_k \in \mathcal{B}$
    while $k \leq ite_{one}$ do
        Solve (3.21). Record the optimal value as $\omega^\star$ and optimal solution as $\boldsymbol{b}'$
        $\boldsymbol{b}_{k+1} \leftarrow \boldsymbol{b}'$
        if $\omega^\star - \omega_k \leq tol_{one}$ then
            break
        end
        $k \leftarrow k+1$, $\omega_k = \omega^\star$
    end
    $\mathcal{D}^{\text{one}} = \{\mathcal{D}^{\text{one}}, \boldsymbol{b}'\}$, $\mathcal{G}^{\text{one}} = \{\mathcal{G}^{\text{one}}, \omega^\star\}$
end
/* Stage-Two Algorithm                                                      */
$\omega^* = \max \mathcal{G}$
if $\lambda_c' > \omega^*$ then
    Define $\mathcal{I}^{\text{two}} = \{i|\omega_i = \arg\max \mathcal{G}^{\text{one}}\}$
    $\mathcal{D}^{\text{two}} = \{\mathcal{D}^{\text{one}}[i]|i \in \mathcal{I}^{\text{two}}\}$
    $\omega = \omega^*$
else
    Define $\mathcal{I}^{\text{two}} = \{i|\omega_i \geq \lambda_c', \omega_i \in \mathcal{G}^{\text{one}}\}$
    $\mathcal{D}^{\text{two}} = [\mathcal{D}^{\text{one}}[i]|i \in \mathcal{I}^{\text{two}}]$
    $\omega = \lambda_c'$
end
$\mathcal{P} = \{\cdot\}$, $\mathcal{Q} = \{\cdot\}$
for $\boldsymbol{b} \in \mathcal{D}^{two}$ do
    $k = 0$, $\boldsymbol{b}_k = \boldsymbol{b}$, $\phi_k = 1e+5$
    while $k \leq ite_{two}$ do
        Solve (3.23). Record the optimal value as $\phi^\star$ and optimal solution as $\boldsymbol{b}'$
        if $\phi_k - \phi^\star \leq tol_{two}$ then
            break
        end
        $k \leftarrow k+1$, $\phi_k = \phi^\star$
    end
    $\mathcal{P} = \{\mathcal{P}, \boldsymbol{b}'\}$, $\mathcal{Q} = \{\mathcal{Q}, \phi^\star\}$
end
$\boldsymbol{b}_{\text{mtd}} = \mathcal{P}[\arg\min_i \mathcal{Q}]$
```

Appendix E

Supplementary Material for E2E-AT: A Unified Framework for Tackling Uncertainty in Task-aware End-to-end Learning

E.1 Network Constrained Economic Dispatch

The stage-one problem is a generator dispatch problem, which can be modelled as:

$$
\begin{aligned}
(\boldsymbol{P}_g^\star, \boldsymbol{\vartheta}^\star, \boldsymbol{s}^\star) \quad & = \arg\min\nolimits_{\boldsymbol{P}_g, \boldsymbol{\vartheta}, \boldsymbol{s}} \boldsymbol{c}_g^T \boldsymbol{P}_g + c_{ls} \mathbf{1}^T \boldsymbol{s} \\
\text{s.t.} \quad & \underline{\boldsymbol{P}}_g \leq \boldsymbol{P}_g \leq \bar{\boldsymbol{P}}_g \\
& \boldsymbol{A}^T \operatorname{diag}(\boldsymbol{b}) \boldsymbol{A} \boldsymbol{\vartheta} = \boldsymbol{C}_g \boldsymbol{P}_g - \boldsymbol{C}_l (\hat{\boldsymbol{y}} - \boldsymbol{s}) \\
& \underline{\boldsymbol{P}}_f \leq \operatorname{diag}(\boldsymbol{b}) \boldsymbol{A} \boldsymbol{\vartheta} \leq \bar{\boldsymbol{P}}_f \\
& \boldsymbol{s} \geq 0 \\
& \boldsymbol{\vartheta}_{\text{ref}} = 0
\end{aligned}
\tag{E.1}
$$

In (E.1), $\boldsymbol{P}_g$ and $\boldsymbol{\vartheta}$ are the vector of generator dispatch and voltage angle in each bus. A linear cost function is considered. All the equality and inequality signs in the constraints are element-wise. The first constraint represents the upper and lower bounds for each generator. The second constraint represents the thermal limit on the power flow in each transmission line where $\boldsymbol{b}$ is the line susceptance, $\boldsymbol{A}$ is the incidence matrix of the power system. The third constraint represents the requirement for power balance on each bus, where $\boldsymbol{C}_g$ and $\boldsymbol{C}_l$ are the incidence matrices of the generator and the load, respectively. $\hat{\boldsymbol{y}}$ represents the forecast load which is parameterized by the neural network model, for example, $\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \boldsymbol{\theta})$. We also add a slack variable $\boldsymbol{s} \geq \mathbf{0}$ as a compensation for infeasibility with large cost coefficient c_{ls}. The fifth constraint indicates that the system is referenced to the slack bus with constant phase angle.

When the generator has been dispatched as $\boldsymbol{P}_g^\star$, we consider a realization on the actual load $\boldsymbol{y}$ by solving the second stage problem (also known as redispatch problem):

$$\begin{aligned}
(\boldsymbol{P}_{ls}^\star, \boldsymbol{P}_{gs}^\star, \boldsymbol{\vartheta}^\star) \quad & = \arg\min\nolimits_{\boldsymbol{P}_{ls}, \boldsymbol{P}_{gs}, \boldsymbol{\vartheta}} \boldsymbol{c}_{ls}^T \boldsymbol{P}_{ls} + \boldsymbol{c}_{gs}^T \boldsymbol{P}_{gs} \\
\text{subject to} \quad & \boldsymbol{A}^T \operatorname{diag}(\boldsymbol{b}) \boldsymbol{A} \boldsymbol{\vartheta} = \boldsymbol{C}_g (\boldsymbol{P}_g^\star - \boldsymbol{P}_{gs}) \\
& \qquad\qquad\qquad\qquad - \boldsymbol{C}_l (\boldsymbol{y} - \boldsymbol{P}_{ls}) \\
& \underline{\boldsymbol{P}}_f \leq \operatorname{diag}(\boldsymbol{b}) \boldsymbol{A} \boldsymbol{\vartheta} \leq \bar{\boldsymbol{P}}_f \\
& \boldsymbol{P}_{ls} \geq 0, \boldsymbol{P}_{gs} \geq 0 \\
& \boldsymbol{\vartheta}_{\text{ref}} = 0
\end{aligned} \tag{E.2}$$

The objective of stage two is to balance the load and to solve any violation of physical constraints of power grid using load shedding $\boldsymbol{P}_{ls}$ if the generator dispatch is lower than the actual load and energy storage $\boldsymbol{P}_{gs}$ if the generator dispatch is higher than the actual load. Since load shedding (similar to blackout) is more critical and should be avoided as much as possible, it is assigned by a larger penalty, i.e. $\boldsymbol{c}_{ls} \gg \boldsymbol{c}_{gs}$.

We highlight that our formulation on power system operation is more realistic than [19], in which the behavior of different loads and network constraints are ignored. This results in more complex COs. In detail, the two COs have more than 60 decision variables and 150 constraints in total, which needs to be exactly solved for every forward pass.

E.2 Certifying the Load Forecasting E2E Learning

Using Proposition 8 and the mixed integer reformulation of NN (4.4), an exact adversarial attack on input $(\boldsymbol{x}, \boldsymbol{y})$ can be found by MILP. It is known that the value of the 'big M' used for NN linearization (e.g. the lower and upper bounds of the output of each NN layer (4.4), the upper bounds of the active inequality dual variables, and the lower bounds of the inactive inequality constraints of the COs) is essential to the performance of MILP solution. Therefore, interval bound propagation (IBP) [153] is used to estimate the layers' bounds in NN. As for COs, we set $M = 10^5$ by experience.

E.2.1 Formulation

By using Proposition 8 and the mixed integer reformulation of NN (4.4), the exact input space adversarial attack on sample $(\boldsymbol{x}, \boldsymbol{y})$ can be found by an MILP:

$$\begin{aligned}
\boldsymbol{\delta}^\star = \max\nolimits_{\boldsymbol{\delta}} \quad & \boldsymbol{c}_g^T \boldsymbol{P}_g + \boldsymbol{c}_{ls}^T \boldsymbol{P}_{ls} + \boldsymbol{c}_{gs}^T \boldsymbol{P}_{gs} \\
\text{subject to} \quad & \boldsymbol{P}_{ls}, \boldsymbol{P}_{gs} \in \mathcal{C}_{\text{Lin-KKT}}^{\text{Reispatch}}(\boldsymbol{P}_{ls}, \boldsymbol{P}_{gs}; \boldsymbol{P}_g) \\
& \boldsymbol{P}_g \in \mathcal{C}_{\text{Lin-KKT}}^{\text{Dispatch}}(\boldsymbol{P}_g; \hat{\boldsymbol{y}}) \\
& \hat{\boldsymbol{y}} \in \mathcal{C}_{\text{nn}}(\boldsymbol{x} + \boldsymbol{\delta}; \boldsymbol{\theta})
\end{aligned} \tag{E.3}$$

where $\mathcal{C}_{\text{nn}}(\cdot)$ is the mixed integer linear representation of the trained neural network by (4.4), $\mathcal{C}_{\text{Lin-KKT}}^{\text{Dispatch}}(\cdot)$ and $\mathcal{C}_{\text{Lin-KKT}}^{\text{Reispatch}}(\cdot)$ are the linearized KKT conditions of the lower level dispatch and redispatch problems, respectively by Proposition 8. Note that (E.3) is a rather simplified representation that omits the detailed formulation of the constraints, the integer variables in $\mathcal{C}_{\text{nn}}(\cdot)$, $\mathcal{C}_{\text{Lin-KKT}}^{\text{Dispatch}}$, and $\mathcal{C}_{\text{Lin-KKT}}^{\text{Reispatch}}$, dual variblaes, as well as the associated lower and upper bounds (big-M) of the integers. Nonetheless, (E.3) is formulated as MILP, which can be solved by solvers like Gurobi.

E.2.2 Interval Bound Propagation (IBP)

Based on (4.4), the lower and upper bounds of each layer in NN can be estimated linearly as:

$$\begin{aligned}\hat{\boldsymbol{l}}_i &= \max\{\mathbf{0}, \boldsymbol{l}_i\} \\ \hat{\boldsymbol{u}}_i &= \max\{\mathbf{0}, \boldsymbol{u}_i\} \\ \boldsymbol{l}_{i+1} &= \max\{\mathbf{0}, \boldsymbol{W}_i\} \cdot \hat{\boldsymbol{l}}_i + \min\{\mathbf{0}, \boldsymbol{W}_i\} \cdot \hat{\boldsymbol{u}}_i + \boldsymbol{b}_i \\ \boldsymbol{u}_{i+1} &= \min\{\mathbf{0}, \boldsymbol{W}_i\} \cdot \hat{\boldsymbol{l}}_i + \max\{\mathbf{0}, \boldsymbol{W}_i\} \cdot \hat{\boldsymbol{u}}_i + \boldsymbol{b}_i\end{aligned} \tag{E.4}$$

for $i = 1, \cdots, d-1$. The initial bound is determined by the attack budget ϵ, that is, $\hat{l}_1 = \boldsymbol{x} - \epsilon \cdot \mathbf{1}$ and $\hat{u}_1 = \boldsymbol{x} + \epsilon \cdot \mathbf{1}$.

E.3 Details Experiment Settings

E.3.1 Data Source

The IEEE bus-14 system is modified from `PyPower`[1].

The meteorological features in the Texas Backbone Power System [164] include temperature (k), longwave radiation (w / m2), shortwave radiation (w / m2), zonal wind speed (m / s), meridional wind speed (m / s) and wind speed (m / s) which are normalized into [0,1]. The calendar feature includes the cosine and sin of the weekday in a week and hour in a day according to their individual period. We pack the meteorological features of 14 buses as well as the 4 calendric features. Therefore, a single datum is $(\boldsymbol{x}^i, \boldsymbol{y}^i) \in \mathbf{R}^{4+6*14} \times \mathbf{R}^{14}$. We map the dataset to the scale that is suitable for the bus-14 system. In detail, we start at small ground-truth load profile and gradually increase to just have the feasible solution of the dispatch and redispatch problems.

[1] `https://github.com/rwl/PYPOWER/blob/master/pypower/case14.py`.

E.3.2 Packages

During inference, we formulate the dispatch and redispatch problem by `Cvxpy` [148], which are solved by calling `Gurobi`[2]. When calculating the gradient, we use `PyTorch` automatic differentiation package and `CvxpyLayers` to implement fast batched forward and backward passes [106].

E.3.3 Network structure

The forecast neural network has three hidden layers with output sizes of 200, 200, and 100. ReLU activations are added between layers. We also add a ReLU activation before the convex layer so that the forecast load is guaranteed to be positive and the adversarial attack cannot result in negative forecast load as well. The ReLU layer is also added when evaluating the certified approach.

E.3.4 Training Settings

For all experiments, we set batch size as 32 and use Adam optimizer [144]. We train the NN with MSE loss for 250 epochs. The learning rate is 10^{-3} with cosine annealing. We store the NN states at 200 epochs and warm-start natural E2E learning for 50 epochs with learning rate 10^{-5}. For E2E-AT, we warm-start training from the state trained by natural E2E learning for 100 epochs with learning rate 10^{-5}. As we set the PGD step to be 7, it is equivalent to 14 epochs under adversarial training for free setting.

For the input space attack, we only attack the meteorological features in the input as the calendric features are discrete and can be easily verified by the operator. For uncertainties in the unpredictable parameter in COs, the susceptance of the transmission line is attacked. This is a realistic setting, as the susceptance of the transmission lines may vary due to temperature changes or can be intentionally altered by the system operator via electronic devices [74]. However, such changes cannot be detected when forecasting the load and during the dispatch stage.

[2] `https://www.gurobi.com`.

Table E.1: Performances of the E2E-AT. The model is trained on a larger dataset and evaluated on the test dataset.

Training Method			**Clean**	**Input Attack**, ϵ_x			**CO Attack**, ϵ_ϕ			**Integrated Attack**, $(\epsilon_x, \epsilon_\phi)$		
ϵ_x	ϵ_ϕ	α	N/A	0.015	0.04	0.06	0.05	0.10	0.15	(0.015,0.05)	(0.04,0.10)	(0.06,0.15)
NAT: Natural Training with Task Loss												
0	0	0	220.3	943.8	2366.2	3528.0	852.8	2254.5	3679.3	1526.4	4287.7	6477.6
AT-MSE: Adversarial Training with MSE Loss												
0.015	N/A	N/A	356.9	836.0	1792.3	2552.5	1294.3	2705.0	4069.1	1727.0	3864.4	5844.9
0.04	N/A	N/A	462.3	745.3	1300.0	1820.6	1292.0	2647.1	4024.5	1558.3	3414.7	5273.2
0.06	N/A	N/A	646.4	846.5	1229.5	1586.8	1352.7	2652.2	4022.7	1540.6	3169.1	4967.4
AT-INPUT: Adversarial Training with Task Loss on the Input Uncertainty												
0.015	0	1.0	302.1	430.3	692.3	981.3	369.2	884.5	1939.9	488.3	1374.5	2981.5
		0.5	240.9	544.4	1322.8	2009.5	756.5	2038.0	3455.0	1074.3	3361.1	5566.8
0.04	0	1.0	429.1	539.4	755.7	950.9	449.5	517.3	656.0	552.1	802.2	1050.6
		0.5	280.9	509.7	972.0	1403.5	739.3	1875.1	3259.2	940.9	2687.7	4647.4
0.06	0	1.0	436.4	528.2	706.5	859.0	449.6	490.7	584.6	538.1	729.9	940.9
		0.5	307.8	432.1	659.9	863.5	553.0	1350.5	2500.5	660.0	1788.9	3268.0
AT-PARA: Adversarial Training with Task Loss on the CO Uncertainty												
0	0.05	1.0	223.0	1003.9	2532.6	3717.9	231.1	967.3	2526.4	1014.9	2908.2	5028.1
		0.5	224.8	1019.5	2538.8	3717.7	239.0	1096.5	2677.8	1028.5	3000.9	5222.0
0	0.10	1.0	227.3	934.2	2456.0	3653.9	227.3	262.7	956.7	949.4	2483.2	3912.2
		0.5	226.9	927.3	2325.5	3392.5	227.1	269.6	1072.2	928.5	2326.8	3760.9
0	0.15	1.0	244.8	1042.6	2552.3	3737.4	244.8	245.5	269.1	1039.3	2545.7	3847.1
		0.5	237.1	974.3	2470.4	3646.2	237.1	237.7	286.6	978.2	2455.3	3733.9
AT-BOTH: Adversarial Training with Task Loss on the Integrated Uncertainties												
0.015	0.05	1.0	305.9	447.8	734.3	1063.7	308.7	389.2	643.5	450.2	816.0	1296.1
		0.5	267.2	743.7	1734.4	2593.4	404.5	1671.6	3150.6	806.5	2865.1	5163.7
0.04	0.10	1.0	385.4	462.6	600.0	731.7	385.8	394.0	438.8	461.4	608.8	782.4
		0.5	293.3	516.2	961.2	1376.6	296.4	456.6	1069.9	516.2	1061.1	1849.6

In addition, we noticed that the objective of E2E adversarial training is nonsmooth. For example, the training objective can be significantly increased when certain inactive CO constraints become active or vice versa. Therefore, we use gradient clip to restrict the 1-norm of the gradient to be maximum 2.0 when updating the network. We found that this setting can result in more stable training.

E.4 Extra Experiment Results

E.4.1 Robust Performance on the Test Dataset

In the main content, we train on 1.0k random samples in the Texas backbone power system and report the clean and robust task-aware cost. Here we train on the entire dataset which contains 8760 samples (i.e., one hour resolution of year 2019). We do random train-test split with proportion 8:2. The training epoch is increased to 200 (or

28 epochs using ‘adversarial training for free’) and the batch size is increased to 64. The remaining training settings are the same as before. We then attack the trained model and report the performance on the **test** dataset.

To observe some new results, we increase the attack budget of the input space uncertainty as $\epsilon_x \in [0.015, 0.04, 0.06]$ to have a task-aware cost comparable to the parameter attack and set the parameter attack budget as $\epsilon_\phi \in [0.05, 0.10, 0.15]$. Furthermore, We report the worst task-aware cost within three multi-runs. Note that we select the worst attack vector for **each** sample in the minibatch.

The extra experiment results are shown in Table E.1. In general terms, the extra experiment in the large data set illustrates similar results to those in Table 4.1. And the main difference is caused by the increase in the input attack budget ϵ_x.

We summarize some of the findings below.

1. In all E2E-AT, the clean accuracy decreases as the attack budget increases, which is consistent to the conventional adversarial training. Moreover, model trained under larger attack budget can also improve robustness under smaller attack budget.

2. Hyperparameter α can balance clean accuracy and robust accuracy, which is consistent with conventional adversarial training.

3. The AT-MSE can impact the robust accuracy under input attack (but still much higher than E2E-AT), but cannot improve the robustness of CO, which actually becomes worse compared to NAT. This is because the AT-MSE only captures the input uncertainties and fits to the MSE loss. Therefore, it cannot be generalized to the uncertainty of CO.

4. Both input and CO uncertainties must be considered when designing the E2E learning task. E2E-AT is an effective approach to improve the model robustness. And the unified training (AT-BOTH) can result in the best task-aware costs in most of the cases.

5. In the new experiment, it can be observed that training under input space adversaries

can improve not only the input robustness but the CO robustness. We actually have opposite results of the experiment in Table 4.1 in which the robustness of CO can improve the robustness of the input. We argue that the main reason is caused by the different influence of the input and CO uncertainties under different attack budgets. This explains why AT-BOTH can significantly improve the robustness of CO but only has a limited improvement on the robustness of input uncertainty. We will discuss this point in the next section.

E.4.2 Gradient Analysis of E2E-AT on the Relationship between Input and CO Uncertainties

Based on the experiment results in Table 4.1 and Table E.1, we speculate that the robustness of one uncertainty can improve the robustness of the other, that is, the two uncertainties are not contradictory in E2E-AT. We thereby give a more detailed discussion.

To start, the two sources of uncertainties, e.g., input and CO uncertainties, are handled by training a **same** robust NN, which is a parametric model to forecast the predictable parameter of CO. According to the sensitivity analysis [157], a robust E2E model under uncertain COs implies that the robust NN can forecast a parameter such that the activations of constraints will not be significantly changed. From the structure of E2E learning, the input uncertainty is amplified by the NN [153], which becomes an additional uncertain parameter of the COs. Similarly, an input-robust E2E model also requires that the NN forecast does not trigger significant constraint violations. Therefore, we speculate that the E2E-AT training objectives under input and CO uncertainties are not contradictory, as they are both reflected and controlled by the behavior of the COs.

However, the effectiveness of AT-INPUT on CO uncertainties (or AT-PARA on input uncertainties) depends on the individual attack budget, which can be analyzed by a gradient analysis. From Table E.1, we compare the uncertainty of the input and unpredictable parameters in CO by designing the attack budget ϵ_x and ϵ_ϕ to have a similar task-aware cost in the clean E2E model. We conclude that the robustness of CO is easier to improve than the robustness of input, and improving the robustness of input can improve the robustness of CO. To verify this point, we calculate the average ℓ_1 norm of the gradient of the NN when input and unpredictable parameter attacks are **separately** generated. Both the NN without E2E-AT (warm started by E2E learning) and the NN updated by AT-BOTH (with $\epsilon_x = 0.04$ and $\epsilon_\phi = 0.10$) are evaluated.

First, as shown in Table E.2, the gradient of E2E-AT is very large and requires a gradient clip during training. Second, although the initial task-aware costs are similar for input and parameter adversaries (2366 vs 2354), the initial gradient under input adversaries is 10 times higher than it of parameter adversaries. Even after AT-BOTH, the difference is still more than 10 times. This implies that the impact of the CO uncertainties on the task-aware cost is much less than the input uncertainties under the current attack budget.

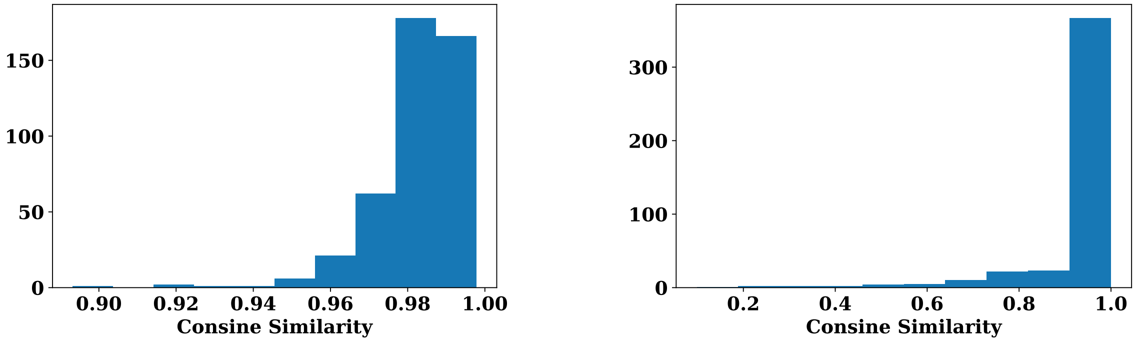
Table E.2: Un-clipped average gradients for E2E-AT in ℓ_1-norm.

	Before E2E-AT	**After E2E-AT**
Input Attack	4.5×10^5	8.4×10^4
CO Attack	5.8×10^4	5.7×10^3

These findings do not violate our initial goal of unifying uncertainties in E2E learning. Actually, we can show that the improvements in the two uncertainties are not contradictory. Otherwise, improving the uncertainties of one cannot improve the uncertainty of the other. To verify the idea, we plot the cosine similarities of the gradients of NN for all mini-batches of size 16 on the entire training dataset. As shown in Fig.E.1, the cosine similarities are close to one, meaning that training on one of the uncertainties can also improve the robustness of the other, which is the same as observed by the experiment results.

Based on the experiment results in Table 4.1 and E.1, as well as the discussions above, we conjecture that, assuming the training objectives on improving the robustness of different uncertainties sources are not contradictory:

1. The gradients of NN under the adversaries/uncertainties from different sources can be used to model their impacts on E2E-AT. It can also give a hint on setting the



(a)

(b)

Figure E.1: Cosine similarities of the gradients of NN under the input space adversarial attack and unpredictable parameter attack in CO. (a): Before E2E-AT (warm start from E2E learning); (b): After E2E-AT.

individual attack budget. And to have more balanced adversarial training behaviors among different uncertainty sources, the attack budgets might be set according to the initial NN gradients other than the task-aware costs.

2. The dominating uncertainty, for example, the one with a higher NN gradient, can dominate the training process and be also effective on the remaining uncertainties. On the contrary, uncertainty with a smaller impact cannot significantly improve the robustness of the others.

Appendix F

Supplementary Material for Task-Aware Machine Unlearning and Its Application in Load Forecasting

F.1 Power System Operation Models

In the US, the following network constrained economic dispatch (NCED) problem is widely adopted [25]. Given the forecast load $\hat{\boldsymbol{y}}$ on each bus,

$$\begin{aligned}
(\boldsymbol{P}_g^\star, \boldsymbol{\vartheta}^\star, \boldsymbol{s}^\star) = \arg\min_{\boldsymbol{P}_g, \boldsymbol{\vartheta}, \boldsymbol{s}} \ & \boldsymbol{P}_g^T \boldsymbol{Q}_g \boldsymbol{P}_g + \boldsymbol{c}_g^T \boldsymbol{P}_g + c_{ls} \|\boldsymbol{s}\|_1 \\
\text{s.t.} \ & \underline{\boldsymbol{P}}_g \leq \boldsymbol{P}_g \leq \bar{\boldsymbol{P}}_g \\
& \boldsymbol{B}_{\text{bus}} \boldsymbol{\vartheta} = \boldsymbol{C}_g \boldsymbol{P}_g - \boldsymbol{C}_l (\hat{\boldsymbol{y}} - \boldsymbol{s}) \\
& \underline{\boldsymbol{P}}_f \leq \boldsymbol{B}_f \boldsymbol{\vartheta} \leq \bar{\boldsymbol{P}}_f \\
& \boldsymbol{s} \geq 0, \quad \boldsymbol{\vartheta}_{\text{ref}} = 0
\end{aligned}$$

In the dispatch problem, a quadratic generator cost is adopted. In addition, $\boldsymbol{B}_{\text{bus}}$ and $\boldsymbol{B}_f$ are the bus susceptance and branch succeptance matrices, respectively. $\boldsymbol{C}_g$ and $\boldsymbol{C}_l$ are the generator and load incidence matrices, respectively. A slack variable $\boldsymbol{s} \geq 0$ with large cost c_{ls} is introduced to ensure feasibility.

After the generators are scheduled, any over- and/or under- generations are penalized when the actual load $\boldsymbol{y}$ is realized in real time. In detail, given $\boldsymbol{P}_g^\star$, we consider the following optimization problem modified from [116]:

$$\begin{aligned}
(\boldsymbol{P}_{ls}^\star, \boldsymbol{P}_{gs}^\star, \boldsymbol{\vartheta}^\star) = \arg \min_{\boldsymbol{P}_{ls}, \boldsymbol{P}_{gs}, \boldsymbol{\vartheta}} & \; c_{ls2} \|\boldsymbol{P}_{ls}\|_2^2 + c_{gs2} \|\boldsymbol{P}_{gs}\|_2^2 \\
& + c_{ls1} \|\boldsymbol{P}_{ls}\|_1 + c_{gs1} \|\boldsymbol{P}_{gs}\|_1 \\
\text{s.t. } & \boldsymbol{B}_{\text{bus}} \boldsymbol{\vartheta} = \boldsymbol{C}_g (\boldsymbol{P}_g^\star - \boldsymbol{P}_{gs}) - \boldsymbol{C}_l (\boldsymbol{y} - \boldsymbol{P}_{ls}) \\
& \underline{\boldsymbol{P}}_f \leq \boldsymbol{B}_f \boldsymbol{\vartheta} \leq \bar{\boldsymbol{P}}_f \\
& \boldsymbol{P}_{ls} \geq 0, \quad \boldsymbol{P}_{gs} \geq 0, \quad \boldsymbol{\vartheta}_{\text{ref}} = 0
\end{aligned}$$

where $\boldsymbol{P}_{ls}$ and $\boldsymbol{P}_{gs}$ are the load shedding and generation storage. The second-order cost $c_{gs2} < c_{ls2}$ and linear cost $c_{gs1} < c_{ls1}$ are set to penalize more on the load shedding.

F.2 Detailed Experiment Settings

F.2.1 Data Description

The meteorological features in the Texas Backbone Power System [164] include temperature (k), long-wave radiation (w / m2), short-wave radiation (w / m2), zonal wind speed (m / s), meridional wind speed (m / s) and wind speed (m / s), which are normalized according to their individual mean and standard deviation. The calendar feature includes the cosine and sin of the weekday in a week and the hour in a day according to their individual period. Therefore, a single datum is $(\boldsymbol{x}^i, \boldsymbol{y}^i) \in \mathbb{R}^{14\times10} \times \mathbb{R}^{14}$. We also normalize the target load by its mean and std. Meanwhile, we use the first 80% data as training dataset and the remaining as test dataset. Finally, the IEEE bus-14 system is modified from `PyPower`.

F.2.2 Linear Load Forecaster

The linear load forecaster can be found by

$$\min_{\boldsymbol{\theta}} \frac{1}{N \cdot 14} \sum_{i=1}^{N} \|\boldsymbol{x}^i \boldsymbol{\theta} - \boldsymbol{y}^i\|_2^2$$

where $\boldsymbol{\theta} \in \mathbb{R}^{10}$. The quadratic objective can be solved analytically or by using conjugate gradient descent.

F.2.3 Convolutional NN Load Forecaster

A CNN is used as feature extractor, which is summarized in Table.F.1.

Table F.1: Structure of the CNN load forecasting model. For the convolutional layer, $(k : w \times h + s + p)$ represents the k number of filters, kernel size $w \times h$ with s stride and p padding in both sides. For the linear layer, the number indicates the output size. The activation function is written in bracket.

Conv Layer 1	8: 3 × 3 + 1 + 1 (ReLU)
Conv Layer 2	8: 4 × 4 + 2 + 1 (ReLU)
Linear Layer 1	64 (tanh)
Linear Layer 2	14 (No activation)

F.2.4 MLP-Mixer Load Forecaster

We also tested the unlearning performance on other NN-based feature extractor, e.g. an MLP-Mixer [200]. Notably, MLP-Mixer only contains two types of multi-layer perceptrons (MLPs), which iteratively capture the information on the feature patches and across the feature patches. In our load forecast setting, it iteratively captures the features within each load and across each load. Regardless of its simple structure, it has been reported that MLP-Mixer can have a performance comparable to CNN or attention-based networks, e.g., transformers [200]. The structure of MLP-Mixer is summarized in Table F.2.

Traing Configuration

We use the same training specifications for CNN and MLP-Mixer. We select the first 30% in the training dataset as the pre-train dataset and the remaining as the user-sensitive

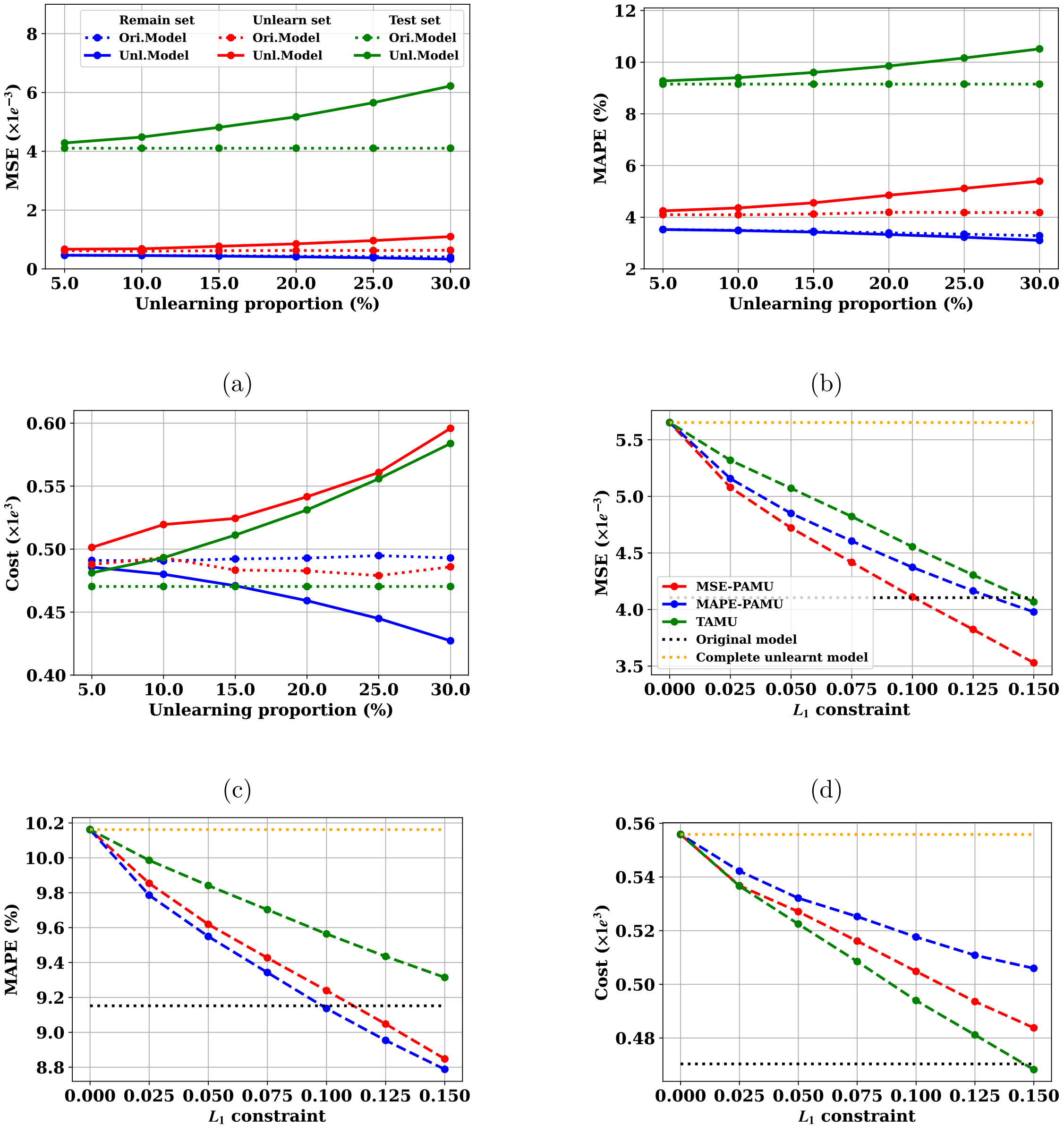
Table F.2: Structure of the MLP-Mixer load forecasting model with exact settings in the original MLP-Mixer paper. One basic Mixer block contains two MLP blocks. Each MLP block contains two linear layers and one activation function between them. We also apply layer norms before each MLP block and pooling layers wherever necessary.

No. of Patches	2
No. of Mixer Blocks	2
MLP	64 (GeLU)
Linear Layer 1	64 (tanh)
Linear Layer 2	14

dataset. The NN forecaster is trained with 100 epochs, batch size of 16, Adam optimizer with learning rate of 10^{-4} and cosine annealing. We also use early stop and record the model with the best performance.

F.3 Extra Experiment Results

The detailed unlearning performances on the CNN and MLP-Mixer based load forecasting models can be found in Fig.F.1 and Fig.F.2, respectively.



(a)

(b)

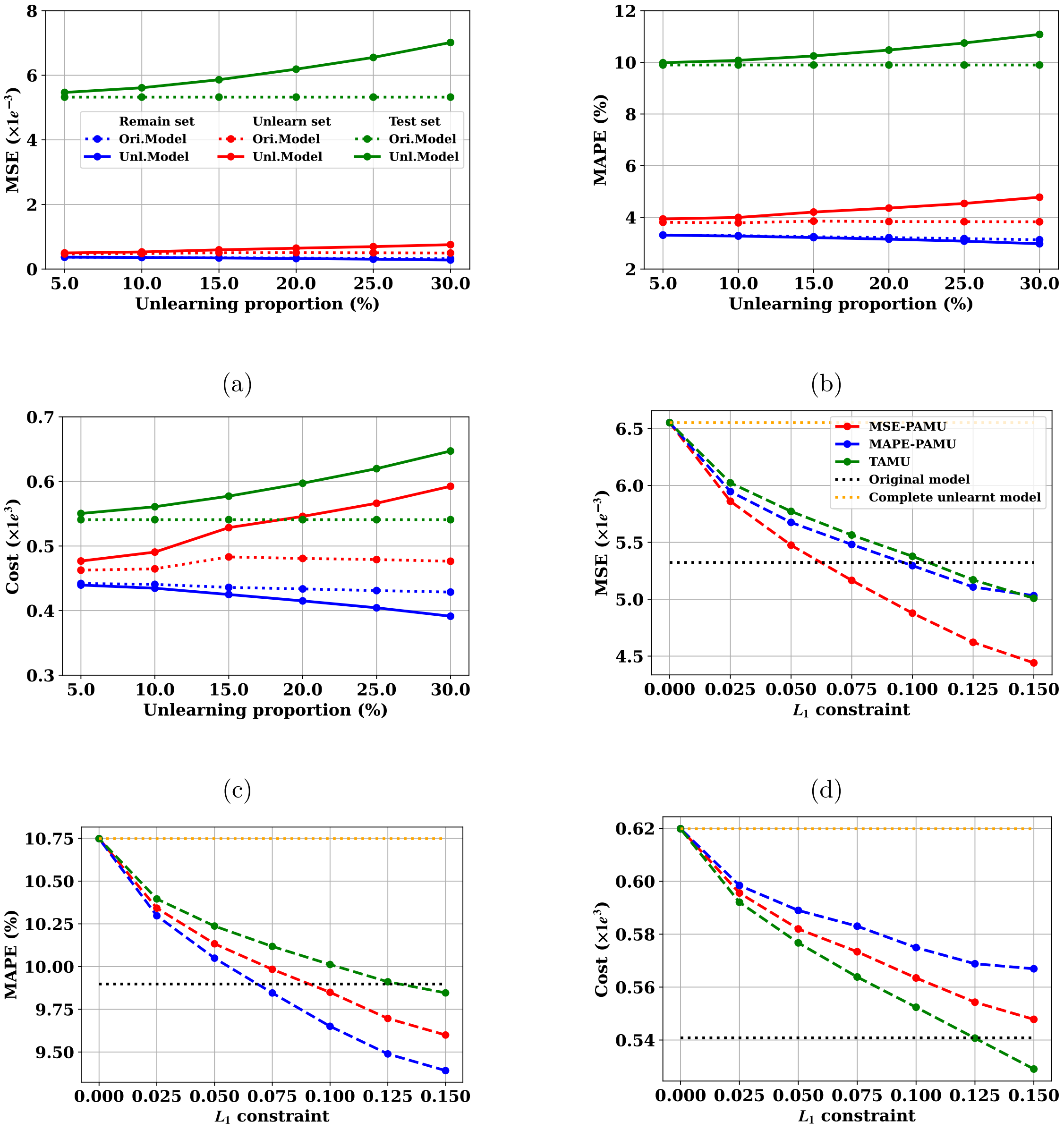
(c)

(d)

(e)

(f)

Figure F.1: Performance on the CNN load forecaster. (a)-(c): performance of complete machine unlearning algorithm (5.8); (d)-(f): Performance of PAMU and TAMU with different test criteria



(a)

(b)

(c)

(d)

(e)

(f)

Figure F.2: Performance on the MLP-Mixer load forecaster. (a)-(c): performance of complete machine unlearning algorithm (5.8); (d)-(f): Performance of PAMU and TAMU with different test criteria

Bibliography

[1] F. Bellizio, W. Xu, D. Qiu, Y. Ye, D. Papadaskalopoulos, J. L. Cremer, F. Teng, and G. Strbac, "Transition to digitalized paradigms for security control and decentralized electricity market," *Proceedings of the IEEE*, vol. 111, no. 7, pp. 744–761, 2023.

[2] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour, "Energy forecasting: A review and outlook," *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 376–388, 2020.

[3] Z. Wang, F. Teng, Y. Zhou, Q. Guo, and H. Sun, "Uncertainty-aware transient stability-constrained preventive redispatch: A distributional reinforcement learning approach," *arXiv preprint arXiv:2402.09263*, 2024.

[4] P. Donti, A. Agarwal, N. V. Bedmutha, L. Pileggi, and J. Z. Kolter, "Adversarially robust learning for security-constrained optimal power flow," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 677–28 689, 2021.

[5] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 630–637.

[6] Z. Wu, Q. Wang, J. Hu, Y. Tang, and Y. Zhang, "Integrating model-driven and data-driven methods for fast state estimation," *International Journal of Electrical Power & Energy Systems*, vol. 139, p. 107982, 2022.

[7] T. Zhang, M. Sun, J. L. Cremer, N. Zhang, G. Strbac, and C. Kang, "A confidence-aware machine learning framework for dynamic security assessment," *IEEE Transactions on Power Systems*, vol. 36, no. 5, pp. 3907–3920, 2021.

[8] Z. Chu, N. Zhang, and F. Teng, "Frequency-constrained resilient scheduling of microgrid: A distributionally robust approach," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 4914–4925, 2021.

[9] F. Bellizio, W. Xu, D. Qiu, Y. Ye, D. Papadaskalopoulos, J. L. Cremer, F. Teng, and G. Strbac, "Transition to digitalized paradigms for security control and decentralized electricity market," *Proceedings of the IEEE*, pp. 1–18, 2022.

[10] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3265–3275, 2018.

[11] C. Ren and Y. Xu, “A fully data-driven method based on generative adversarial networks for power system dynamic security assessment with missing data,” *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 5044–5052, 2019.

[12] H. Shengren, P. P. Vergara, E. M. S. Duque, and P. Palensky, “Optimal energy system scheduling using a constraint-aware reinforcement learning algorithm,” *International Journal of Electrical Power & Energy Systems*, vol. 152, p. 109230, 2023.

[13] D. Qiu, Y. Ye, D. Papadaskalopoulos, and G. Strbac, “Scalable coordinated management of peer-to-peer energy trading: A multi-cluster deep reinforcement learning approach,” *Applied energy*, vol. 292, p. 116940, 2021.

[14] J. Wang, W. Xu, Y. Gu, W. Song, and T. Green, “Multi-agent reinforcement learning for active voltage control on power distribution networks,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *arXiv preprint arXiv:1312.6199*, 2013.

[16] U. Sadana, A. Chenreddy, E. Delage, A. Forel, E. Frejinger, and T. Vidal, “A survey of contextual optimization methods for decision making under uncertainty,” *arXiv preprint arXiv:2306.10374*, 2023.

[17] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[18] Y. Bengio, “Using a financial training criterion rather than a prediction criterion,” *International journal of neural systems*, vol. 8, no. 04, pp. 433–443, 1997.

[19] P. Donti, B. Amos, and J. Z. Kolter, “Task-based end-to-end model learning in stochastic optimization,” *Advances in neural information processing systems*, vol. 30, 2017.

[20] W. Xu, J. Wang, and F. Teng, “E2e-at: A unified framework for tackling uncertainty in task-aware end-to-end learning,” *arXiv preprint arXiv:2312.10587, acceped by AAAI-24*, 2023.

[21] A. N. Elmachtoub and P. Grigas, “Smart “predict, then optimize” ,” *Management Science*, vol. 68, no. 1, pp. 9–26, 2022.

[22] B. Wilder, B. Dilkina, and M. Tambe, “Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1658–1665.

[23] J. Mandi, J. Kotary, S. Berden, M. Mulamba, V. Bucarey, T. Guns, and F. Fioretto, “Decision-focused learning: Foundations, state of the art, benchmark and future opportunities,” *arXiv preprint arXiv:2307.13565*, 2023.

[24] J. Kotary, V. Di Vito, J. Christopher, P. Van Hentenryck, and F. Fioretto, "Predict-then-optimize by proxy: Learning joint models of prediction and optimization," *arXiv preprint arXiv:2311.13087*, 2023.

[25] A. J. Conejo and L. Baringo, *Power system operations.* Springer, 2018, vol. 11.

[26] M. Sun, J. Cremer, and G. Strbac, "A novel data-driven scenario generation framework for transmission expansion planning with high renewable energy penetration," *Applied energy*, vol. 228, pp. 546–555, 2018.

[27] M. Liu, Z. Zhang, P. Ge, R. Deng, M. Sun, J. Chen, and F. Teng, "Enhancing cyber-resiliency of der-based smartgrid: A survey," *arXiv preprint arXiv:2305.05338*, 2023.

[28] X. Wan, M. Sun, B. Chen, Z. Chu, and F. Teng, "Adapsafe: adaptive and safe-certified deep reinforcement learning-based frequency control for carbon-neutral power systems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 5294–5302.

[29] C.-W. Ten, C.-C. Liu, and G. Manimaran, "Vulnerability assessment of cybersecurity for scada systems," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1836–1846, 2008.

[30] R. V. Yohanandhan, R. M. Elavarasan, R. Pugazhendhi, M. Premkumar, L. Mihet-Popa, and V. Terzija, "A holistic review on cyber-physical power system (cpps) testbeds for secure and sustainable electric power grid–part–i: Background on cpps and necessity of cpps testbeds," *International Journal of Electrical Power & Energy Systems*, vol. 136, p. 107718, 2022.

[31] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2218–2234, 2019.

[32] R. M. Lee, M. Assante, and T. J. Conway, "Analysis of the cyber attack on the ukrainian power grid," *Electricity Information Sharing and Analysis Center (E-ISAC)*, vol. 388, 2016.

[33] J. Bi, F. Luo, S. He, G. Liang, W. Meng, and M. Sun, "False data injection-and propagation-aware game theoretical approach for microgrids," *IEEE Transactions on Smart Grid*, 2022.

[34] S. Tan, P. Xie, J. M. Guerrero, and J. C. Vasquez, "False data injection cyber-attacks detection for multiple dc microgrid clusters," *Applied Energy*, vol. 310, p. 118425, 2022.

[35] S. Gao, J. Lei, X. Wei, Y. Liu, and T. Wang, "A novel bilevel false data injection attack model based on pre-and post-dispatch," *IEEE Transactions on Smart Grid*, pp. 1–1, 2022.

[36] C. Chen, M. Cui, X. Fang, B. Ren, and Y. Chen, "Load altering attack-tolerant defense strategy for load frequency control system," *Applied Energy*, vol. 280, p. 116015, 2020.

[37] D. Choeum and D.-H. Choi, “Trilevel smart meter hardening strategy for mitigating cyber attacks against volt/var optimization in smart power distribution systems,” *Applied Energy*, vol. 304, p. 117710, 2021.

[38] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation.* CRC press, 2004.

[39] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, “Malicious data attacks on the smart grid,” *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.

[40] X. Liu, Z. Li, X. Liu, and Z. Li, “Masking transmission line outages via false data injection attacks,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1592–1602, 2016.

[41] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, pp. 1–33, 2011.

[42] G. Hug and J. A. Giampapa, “Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks,” *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1362–1370, 2012.

[43] M. A. Rahman and H. Mohsenian-Rad, “False data injection attacks against nonlinear state estimation in smart power grids,” in *2013 IEEE Power & Energy Society General Meeting.* IEEE, 2013, pp. 1–5.

[44] S. Ahmed, Y. Lee, S.-H. Hyun, and I. Koo, “Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2765–2777, 2019.

[45] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, “On false data-injection attacks against power system state estimation: Modeling and countermeasures,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 717–729, 2014.

[46] K. Lai, M. Illindala, and K. Subramaniam, “A tri-level optimization model to mitigate coordinated attacks on electric power systems in a cyber-physical environment,” *Applied energy*, vol. 235, pp. 204–218, 2019.

[47] M. N. Kurt, Y. Yılmaz, and X. Wang, “Real-time detection of hybrid and stealthy cyber-attacks in smart grid,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 498–513, 2018.

[48] K. Manandhar, X. Cao, F. Hu, and Y. Liu, “Detection of faults and attacks including false data injection attack in smart grid using kalman filter,” *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 370–379, 2014.

[49] Z. Chu, O. Kosut, and L. Sankar, “Detecting load redistribution attacks via support vector models,” *arXiv preprint arXiv:2003.06543*, 2020.

[50] M. Cui, J. Wang, and B. Chen, “Flexible machine learning-based cyberattack detection using spatiotemporal patterns for distribution systems,” *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1805–1808, 2020.

[51] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, and S. Mishra, “Decision tree and svm-based data analytics for theft detection in smart grid,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1005–1016, 2016.

[52] T. Wu, W. Xue, H. Wang, C. Y. Chung, G. Wang, J. Peng, and Q. Yang, “Extreme learning machine-based state reconstruction for automatic attack filtering in cyber physical power system,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1892–1904, 2021.

[53] S. Ahmed, Y. Lee, S. Hyun, and I. Koo, “Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2765–2777, 2019.

[54] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor, “Machine learning methods for attack detection in the smart grid,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1773–1786, 2016.

[55] C. Wang, S. Tindemans, K. Pan, and P. Palensky, “Detection of false data injection attacks using the autoencoder approach,” *arXiv preprint arXiv:2003.02229*, 2020.

[56] J. Zhao, G. Zhang, M. La Scala, Z. Y. Dong, C. Chen, and J. Wang, “Short-term state forecasting-aided method for detection of smart grid general false data injection attacks,” *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1580–1590, 2017.

[57] A. Ashok, M. Govindarasu, and V. Ajjarapu, “Online detection of stealthy false data injection attacks in power system state estimation,” *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1636–1646, 2018.

[58] K. Pan, P. Palensky, and P. M. Esfahani, “Dynamic anomaly detection with high-fidelity simulators: A convex optimization approach,” *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1500–1515, 2022.

[59] M. Higgins, K. Mayes, and F. Teng, “Enhanced cyber-physical security using attack-resistant cyber nodes and event-triggered moving target defence,” *arXiv preprint arXiv:2010.14173*, 2020.

[60] X. Lei, Z. Yang, J. Yu, J. Zhao, Q. Gao, and H. Yu, “Data-driven optimal power flow: A physics-informed machine learning approach,” *IEEE Transactions on Power Systems*, vol. 36, no. 1, pp. 346–354, 2020.

[61] J. Kim, L. Tong, and R. J. Thomas, “Subspace methods for data attack on state estimation: A data driven approach,” *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1102–1114, 2014.

[62] Z.-H. Yu and W.-L. Chin, "Blind false data injection attack using pca approximation method in smart grid," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1219–1226, 2015.

[63] S. Lakshminarayana, A. Kammoun, M. Debbah, and H. V. Poor, "Data-driven false data injection attacks against power grids: A random matrix approach," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 635–646, 2021.

[64] J.-H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 709–745, 2020.

[65] K. L. Morrow, E. Heine, K. M. Rogers, R. B. Bobba, and T. J. Overbye, "Topology perturbation for detecting malicious data injection," in *2012 45th Hawaii International Conference on System Sciences*, 2012, pp. 2104–2113.

[66] K. R. Davis, K. L. Morrow, R. Bobba, and E. Heine, "Power flow cyber attacks and perturbation-based defense," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, 2012, pp. 342–347.

[67] M. A. Rahman, E. Al-Shaer, and R. B. Bobba, "Moving target defense for hardening the security of the power system state estimation," in *Proceedings of the First ACM Workshop on Moving Target Defense*, 2014, pp. 59–68.

[68] C. Liu, J. Wu, C. Long, and D. Kundur, "Reactance perturbation for detecting and identifying fdi attacks in power system state estimation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 763–776, 2018.

[69] Z. Zhang, R. Deng, D. K. Yau, P. Cheng, and J. Chen, "Analysis of moving target defense against false data injection attacks on power grid," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2320–2335, 2019.

[70] B. Liu and H. Wu, "Optimal d-facts placement in moving target defense against false data injection attacks," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4345–4357, 2020.

[71] Z. Zhang, R. Deng, P. Cheng, and M.-Y. Chow, "Strategic protection against fdi attacks with moving target defense in power grids," *IEEE Transactions on Control of Network Systems*, pp. 1–1, 2021.

[72] M. Liu, C. Zhao, Z. Zhang, and R. Deng, "Explicit analysis on effectiveness and hiddenness of moving target defense in ac power systems," *IEEE Transactions on Power Systems*, pp. 1–1, 2022.

[73] S. Lakshminarayana and D. K. Yau, "Cost-benefit analysis of moving-target defense in power grids," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1152–1163, 2021.

[74] W. Xu, I. M. Jaimoukh, and F. Teng, "Robust moving target defence against false data injection attacks in power grids," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2022.

[75] J. Tian, R. Tan, X. Guan, and T. Liu, "Enhanced hidden moving target defense in smart grids," *IEEE transactions on smart grid*, vol. 10, no. 2, pp. 2208–2223, 2018.

[76] Z. Zhang, R. Deng, D. K. Yau, P. Cheng, and J. Chen, "On hiddenness of moving target defense against false data injection attacks on power grid," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 3, pp. 1–29, 2020.

[77] B. Liu and H. Wu, "Optimal planning and operation of hidden moving target defense for maximal detection effectiveness," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4447–4459, 2021.

[78] M. Higgins, F. Teng, and T. Parisini, "Stealthy mtd against unsupervised learning-based blind fdi attacks in power systems," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1275–1287, 2020.

[79] W. Xu, M. Higgins, J. Wang, I. M. Jaimoukha, and F. Teng, "Blending data and physics against false data injection attack: An event-triggered moving target defence approach," *IEEE Transactions on Smart Grid*, vol. 14, no. 4, pp. 3176–3188, 2023.

[80] C. Liu, H. Liang, T. Chen, J. Wu, and C. Long, "Joint admittance perturbation and meter protection for mitigating stealthy fdi attacks against power system state estimation," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1468–1478, 2019.

[81] C. Liu, R. Deng, W. He, H. Liang, and W. Du, "Optimal coding schemes for detecting false data injection attacks in power system state estimation," *IEEE Transactions on Smart Grid*, vol. 13, no. 1, pp. 738–749, 2021.

[82] J. Tian, R. Tan, X. Guan, Z. Xu, and T. Liu, "Moving target defense approach to detecting stuxnet-like attacks," *IEEE transactions on smart grid*, vol. 11, no. 1, pp. 291–300, 2019.

[83] M. Liu, C. Zhao, Z. Zhang, R. Deng, P. Cheng, and J. Chen, "Converter-based moving target defense against deception attacks in dc microgrids," *IEEE Transactions on Smart Grid*, 2021.

[84] M. Liu, C. Zhao, Z. Zhang, R. Deng, and P. Cheng, "Analysis of moving target defense in unbalanced and multiphase distribution systems considering voltage stability," in *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2021, pp. 207–213.

[85] H. Zhang, B. Liu, X. Liu, A. Pahwa, and H. Wu, "Voltage stability constrained moving target defense against net load redistribution attacks," *IEEE Transactions on Smart Grid*, vol. 13, no. 5, pp. 3748–3759, 2022.

[86] H. Zhang, N. Fulk, B. Liu, L. Edmonds, X. Liu, and H. Wu, "Load margin constrained moving target defense against false data injection attacks," in *2022 IEEE Green Technologies Conference (GreenTech)*. IEEE, 2022, pp. 51–56.

[87] M. Higgins, W. Xu, F. Teng, and T. Parisini, "Cyber-physical risk assessment for false data injection attacks considering moving target defences," *International Journal of Information Security*, 2022.

[88] C. Liu, M. Zhou, J. Wu, C. Long, A. Farraj, E. Hammad, and D. Kundur, "Reactance perturbation for enhancing detection of fdi attacks in power system state estimation," in *2017 IEEE Global conference on signal and information processing (GlobalSIP)*. IEEE, 2017, pp. 523–527.

[89] S. Lakshminarayana, E. V. Belmega, and H. V. Poor, "Moving-target defense for detecting coordinated cyber-physical attacks in power grids," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2019, pp. 1–7.

[90] ——, "Moving-target defense against cyber-physical attacks in power grids via game theory," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 5244–5257, 2021.

[91] Z. Zhang, R. Deng, D. Yau, P. Cheng, and J. Chen, "On effectiveness of detecting fdi attacks on power grid using moving target defense," in *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2019, pp. 1–5.

[92] C. Liu, H. Liang, T. Chen, J. Wu, and C. Long, "Joint admittance perturbation and meter protection for mitigating stealthy fdi attacks against power system state estimation," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1468–1478, 2020.

[93] Z. Zhang, R. Deng, D. K. Y. Yau, P. Cheng, and J. Chen, "Analysis of moving target defense against false data injection attacks on power grid," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2320–2335, 2020.

[94] B. Liu and H. Wu, "Systematic planning of moving target defence for maximising detection effectiveness against false data injection attacks in smart grid," *IET Cyber-Physical Systems: Theory & Applications*, vol. 6, no. 3, pp. 151–163, 2021.

[95] B. Liu, H. Wu, A. Pahwa, F. Ding, E. Ibrahim, and T. Liu, "Hidden moving target defense against false data injection in distribution network reconfiguration," in *2018 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2018, pp. 1–5.

[96] Z. Zhang, Y. Tian, R. Deng, and J. Ma, "A double-benefit moving target defense against cyber-physical attacks in smart grid," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[97] Y. Deng and S. Sen, "Predictive stochastic programming," *Computational Management Science*, pp. 1–34, 2022.

[98] R. Chen and I. Paschalidis, “Selecting optimal decisions via distributionally robust nearest-neighbor regression,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[99] J. M. Morales, M. Muñoz, and S. Pineda, “Prescribing net demand for two-stage electricity generation scheduling,” *Operations Research Perspectives*, vol. 10, p. 100268, 2023.

[100] X. Chen, Y. Liu, and L. Wu, “Improving electricity market economy via closed-loop predict-and-optimize,” *arXiv preprint arXiv:2208.13065*, 2022.

[101] B. Tang and E. B. Khalil, “Pyepo: A pytorch-based end-to-end predict-then-optimize library for linear and integer programming,” *arXiv preprint arXiv:2206.14234*, 2022.

[102] J. Domke, “Generic methods for optimization-based modeling,” in *Artificial Intelligence and Statistics.* PMLR, 2012, pp. 318–326.

[103] Y. Lu, Z. Li, Y. Zhou, N. Li, and Y. Mo, “Bridging the gaps: Learning verifiable model-free quadratic programming controllers inspired by model predictive control,” *arXiv preprint arXiv:2312.05332*, 2023.

[104] S. G. Krantz and H. R. Parks, *The implicit function theorem: history, theory, and applications.* Springer Science & Business Media, 2002.

[105] B. Amos and J. Z. Kolter, “Optnet: Differentiable optimization as a layer in neural networks,” in *International Conference on Machine Learning.* PMLR, 2017, pp. 136–145.

[106] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, “Differentiable convex optimization layers,” *Advances in neural information processing systems*, vol. 32, 2019.

[107] A. Ferber, B. Wilder, B. Dilkina, and M. Tambe, “Mipaal: Mixed integer program as a layer,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 02, 2020, pp. 1504–1511.

[108] G. Li and H.-D. Chiang, “Toward cost-oriented forecasting of wind power generation,” *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2508–2517, 2018.

[109] T. Carriere and G. Kariniotakis, “An integrated approach for value-oriented energy forecasting and data-driven decision-making application to renewable energy trading,” *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6933–6944, 2019.

[110] M. A. Muñoz, J. M. Morales, and S. Pineda, “Feature-driven improvement of renewable energy forecasting and trading,” *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 3753–3763, 2020.

[111] J. Han, L. Yan, and Z. Li, “A task-based day-ahead load forecasting model for stochastic economic dispatch,” *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5294–5304, 2021.

[112] J. D. Garcia, A. Street, T. Homem-de Mello, and F. D. Munoz, "Application-driven learning via joint prediction and optimization of demand and reserves requirement," *arXiv preprint arXiv:2102.13273*, 2021.

[113] C. Zhao, C. Wan, and Y. Song, "Operating reserve quantification using prediction intervals of wind power: An integrated probabilistic forecasting and decision methodology," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3701–3714, 2021.

[114] X. Chen, Y. Yang, Y. Liu, and L. Wu, "Feature-driven economic improvement for network-constrained unit commitment: A closed-loop predict-and-optimize framework," *IEEE Transactions on Power Systems*, vol. 37, no. 4, pp. 3104–3118, 2022.

[115] C. Zhao, C. Wan, and Y. Song, "Cost-oriented prediction intervals: On bridging the gap between forecasting and decision," *IEEE Transactions on Power Systems*, vol. 37, no. 4, pp. 3048–3062, 2022.

[116] J. Zhang, Y. Wang, and G. Hug, "Cost-oriented load forecasting," *Electric Power Systems Research*, vol. 205, p. 107723, 2022.

[117] L. Sang, Y. Xu, H. Long, Q. Hu, and H. Sun, "Electricity price prediction for energy storage system arbitrage: A decision-focused approach," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 2822–2832, 2022.

[118] C. Lu, W. Jiang, and C. Wu, "Effective end-to-end learning framework for economic dispatch," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2673–2683, 2022.

[119] A. Stratigakos, S. Camal, A. Michiorri, and G. Kariniotakis, "Prescriptive trees for integrated forecasting and optimization applied in trading of renewable energy," *IEEE Transactions on Power Systems*, vol. 37, no. 6, pp. 4696–4708, 2022.

[120] H. Zhang, R. Li, Y. Chen, Z. Chu, M. Sun, and F. Teng, "Risk-aware objective-based forecasting in inertia management," *arXiv preprint arXiv:2303.11785*, 2023.

[121] Y. Zhou, Q. Wen, J. Song, X. Cui, and Y. Wang, "Load data valuation in multi-energy systems: An end-to-end approach," *arXiv preprint arXiv:2311.09839*, 2023.

[122] D. Wahdany, C. Schmitt, and J. L. Cremer, "More than accuracy: end-to-end wind power forecasting that optimises the energy system," *Electric Power Systems Research*, vol. 221, p. 109384, 2023.

[123] R. Vohra, A. Rajaei, and J. L. Cremer, "End-to-end learning with multiple modalities for system-optimised renewables nowcasting," *arXiv preprint arXiv:2304.07151*, 2023.

[124] H. Zhang, R. Li, M. Sun, and T. Fei, "Adaptive decision-objective loss for forecast-then-optimize in power systems," *arXiv preprint arXiv:2312.13501*, 2023.

[125] Y. Zhang, H. Wen, and Q. Wu, "A contextual bandit approach for value-oriented prediction interval forecasting," *IEEE Transactions on Smart Grid*, 2023.

[126] W. Xu and F. Teng, "Task-aware machine unlearning and its application in load forecasting," *arXiv preprint arXiv:2308.14412*, 2023.

[127] V. Dvorkin, S. Delikaraoglou, and J. M. Morales, "Setting reserve requirements to approximate the efficiency of the stochastic dispatch," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1524–1536, 2019.

[128] N. Viafora, S. Delikaraoglou, P. Pinson, G. Hug, and J. Holbøll, "Dynamic reserve and transmission capacity allocation in wind-dominated power systems," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3017–3028, 2021.

[129] A. Gomez-Exposito, A. J. Conejo, and C. Cañizares, *Electric energy systems: analysis and operation.* CRC press, 2018.

[130] J. Zhang, Y. Wang, Y. Weng, and N. Zhang, "Topology identification and line parameter estimation for non-pmu distribution network: A numerical method," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4440–4453, 2020.

[131] W. Xu and F. Teng, "A deep learning based detection method for combined integrity-availability cyber attacks in power system," *arXiv preprint arXiv:2011.01816*, 2020.

[132] B. Li, G. Xiao, R. Lu, R. Deng, and H. Bao, "On feasibility and limitations of detecting false data injection attacks on power grid state estimation using d-facts devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 854–864, 2019.

[133] K. Krishnamoorthy, *Handbook of statistical distributions with applications.* Chapman and Hall/CRC, 2006.

[134] A. Teixeira, S. Amin, H. Sandberg, K. H. Johansson, and S. S. Sastry, "Cyber security analysis of state estimators in electric power systems," in *49th IEEE conference on decision and control (CDC).* IEEE, 2010, pp. 5991–5998.

[135] C. D. Meyer, *Matrix analysis and applied linear algebra.* Siam, 2000, vol. 71.

[136] A. Galántai, "Subspaces, angles and pairs of orthogonal projections," *Linear and Multilinear Algebra*, vol. 56, no. 3, pp. 227–260, 2008.

[137] Z. Galil, "Efficient algorithms for finding maximum matching in graphs," *ACM Computing Surveys (CSUR)*, vol. 18, no. 1, pp. 23–38, 1986.

[138] J. A. Bondy and U. S. R. Murty, *Graph theory.* Springer, 2008, vol. 244.

[139] H. Zhu and G. B. Giannakis, "Power system nonlinear state estimation using distributed semidefinite programming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1039–1050, 2014.

[140] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[141] R. Zimmerman and C. MurilloSanchez, “Matpower user’ s manual version 7.0 b1,” *Power Systems Engineering Research Center (PSerc): Tempe, AZ, USA*, 2018.

[142] M. Higgins, J. Zhang, N. Zhang, and F. Teng, “Topology learning aided false data injection attack without prior topology information,” in *2021 IEEE Power Energy Society General Meeting (PESGM)*, 2021, pp. 1–5.

[143] H. Zhang, B. Liu, and H. Wu, “Smart grid cyber-physical attack and defense: A review,” *IEEE Access*, vol. 9, pp. 29 641–29 659, 2021.

[144] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[145] B. Liu, Q. Yang, H. Zhang, and H. Wu, “An interior-point solver for ac optimal power flow considering variable impedance-based facts devices,” *IEEE Access*, vol. 9, pp. 154 460–154 470, 2021.

[146] A. Sinha, P. Malo, and K. Deb, “A review on bilevel optimization: from classical to evolutionary approaches and applications,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 276–295, 2017.

[147] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.

[148] S. Diamond and S. Boyd, “Cvxpy: A python-embedded modeling language for convex optimization,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[149] B. Chen, P. L. Donti, K. Baker, J. Z. Kolter, and M. Bergés, “Enforcing policy feasibility constraints through differentiable projection for energy optimization,” in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, 2021, pp. 199–210.

[150] Z. Liu, Y. Yin, F. Bai, and D. K. Grimm, “End-to-end learning of user equilibrium with implicit neural networks,” *Transportation Research Part C: Emerging Technologies*, vol. 150, p. 104085, 2023.

[151] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” *arXiv preprint arXiv:1706.06083*, 2017.

[152] V. Tjeng, K. Xiao, and R. Tedrake, “Evaluating robustness of neural networks with mixed integer programming,” *arXiv preprint arXiv:1711.07356*, 2017.

[153] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli, “On the effectiveness of interval bound propagation for training verifiably robust models,” *arXiv preprint arXiv:1810.12715*, 2018.

[154] W. Xu and F. Teng, "Availability adversarial attack and countermeasures for deep learning-based load forecasting," in *2023 IEEE Belgrade PowerTech*, 2023, pp. 01–06.

[155] J. Kotary, F. Fioretto, P. Van Hentenryck, and B. Wilder, "End-to-end constrained optimization learning: A survey," *arXiv preprint arXiv:2103.16378*, 2021.

[156] L. Kong, J. Cui, Y. Zhuang, R. Feng, B. A. Prakash, and C. Zhang, "End-to-end stochastic optimization with energy-based model," *Advances in Neural Information Processing Systems*, vol. 35, pp. 11 341–11 354, 2022.

[157] S. P. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[158] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical programs with equilibrium constraints.* Cambridge University Press, 1996.

[159] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning.* PMLR, 2019, pp. 7472–7482.

[160] Y. Dong, Z. Deng, T. Pang, J. Zhu, and H. Su, "Adversarial distributional training for robust deep learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8270–8283, 2020.

[161] K. Y. Xiao, V. Tjeng, N. M. Shafiullah, and A. Madry, "Training for faster adversarial robustness verification via inducing relu stability," *arXiv preprint arXiv:1809.03008*, 2018.

[162] A. Agarwal, P. L. Donti, J. Z. Kolter, and L. Pileggi, "Employing adversarial robustness techniques for large-scale stochastic optimal power flow," *Electric Power Systems Research*, vol. 212, p. 108497, 2022.

[163] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[164] J. Lu, X. Li, H. Li, T. Chegini, C. Gamarra, Y. Yang, M. Cook, and G. Dillingham, "A synthetic texas backbone power system with climate-dependent spatio-temporal correlated profiles," *arXiv preprint arXiv:2302.13231*, 2023.

[165] J. Xie, T. Hong, and J. Stroud, "Long-term retail energy forecasting with consideration of residential customer attrition," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2245–2252, 2015.

[166] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, 2016.

[167] E. Ebeid, R. Heick, and R. H. Jacobsen, "Deducing energy consumer behavior from smart meter data," *Future Internet*, vol. 9, no. 3, p. 29, 2017.

[168] J. Luo, T. Hong, and S.-C. Fang, "Benchmarking robustness of load forecasting models under data integrity attacks," *International Journal of Forecasting*, vol. 34, no. 1, pp. 89–104, 2018.

[169] Y. Liang, D. He, and D. Chen, "Poisoning attack on load forecasting," in *2019 IEEE innovative smart grid technologies-Asia (ISGT Asia)*. IEEE, 2019, pp. 1230–1235.

[170] Y. Wang, N. Gao, and G. Hug, "Personalized federated learning for individual consumer load forecasting," *CSEE Journal of Power and Energy Systems*, 2022.

[171] Y. Dong, Y. Chen, X. Zhao, and X. Huang, "Short-term load forecasting with distributed long short-term memory," *arXiv preprint arXiv:2208.01147*, 2022.

[172] E. U. Soykan, Z. Bilgin, M. A. Ersoy, and E. Tomur, "Differentially private deep learning for load forecasting on smart grid," in *2019 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2019, pp. 1–6.

[173] J. D. Fernández, S. P. Menci, C. M. Lee, A. Rieger, and G. Fridgen, "Privacy-preserving federated learning for residential short-term load forecasting," *Applied Energy*, vol. 326, p. 119915, 2022.

[174] M. A. Husnoo, A. Anwar, N. Hosseinzadeh, S. N. Islam, A. N. Mahmood, and R. Doss, "A secure federated learning framework for residential short term load forecasting," *IEEE Transactions on Smart Grid*, 2023.

[175] A. Mantelero, "The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'," *Computer Law & Security Review*, vol. 29, no. 3, pp. 229–235, 2013.

[176] T. Shaik, X. Tao, H. Xie, L. Li, X. Zhu, and Q. Li, "Exploring the landscape of machine unlearning: A survey and taxonomy," *arXiv preprint arXiv:2305.06360*, 2023.

[177] C. Yu, S. Jeoung, A. Kasi, P. Yu, and H. Ji, "Unlearning bias in language models by partitioning gradients," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 6032–6048.

[178] Y. Zeng, J. Xu, Y. Li, C. Chen, Q. Dai, and Z. Du, "Towards highly-efficient and accurate services qos prediction via machine unlearning," *IEEE Access*, 2023.

[179] H. Xia, S. Xu, J. Pei, R. Zhang, Z. Yu, W. Zou, L. Wang, and C. Liu, "Fedme2: Memory evaluation & erase promoting federated unlearning in dtmn," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3573–3588, 2023.

[180] Z. Zhang, M. Tian, C. Li, Y. Huang, and L. Yang, "Poison neural network-based mmwave beam selection and detoxification with machine unlearning," *IEEE Transactions on Communications*, vol. 71, no. 2, pp. 877–892, 2023.

[181] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 463–480.

[182] A. Ginart, M. Guan, G. Valiant, and J. Y. Zou, "Making ai forget you: Data deletion in machine learning," *Advances in neural information processing systems*, vol. 32, 2019.

[183] J. Brophy and D. Lowd, "Machine unlearning for random forests," in *International Conference on Machine Learning.* PMLR, 2021, pp. 1092–1104.

[184] A. Golatkar, A. Achille, and S. Soatto, "Eternal sunshine of the spotless net: Selective forgetting in deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9304–9312.

[185] A. Peste, D. Alistarh, and C. H. Lampert, "Ssse: Efficiently erasing samples from trained machine learning models," *arXiv preprint arXiv:2107.03860*, 2021.

[186] S. Fu, F. He, Y. Xu, and D. Tao, "Bayesian inference forgetting," *arXiv preprint arXiv:2101.06417*, 2021.

[187] A. Golatkar, A. Achille, A. Ravichandran, M. Polito, and S. Soatto, "Mixed-privacy forgetting in deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 792–801.

[188] C. Guo, T. Goldstein, A. Hannun, and L. Van Der Maaten, "Certified data removal from machine learning models," *arXiv preprint arXiv:1911.03030*, 2019.

[189] J. Bae, N. Ng, A. Lo, M. Ghassemi, and R. B. Grosse, "If influence functions are the answer, then what is the question?" *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 953–17 967, 2022.

[190] T. Hoang, S. Rana, S. Gupta, and S. Venkatesh, "Learn to unlearn for deep neural networks: Minimizing unlearning interference with gradient projection," in *WACV*, Jan 2024.

[191] L. Graves, V. Nagisetty, and V. Ganesh, "Amnesiac machine learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, 2021, pp. 11 516–11 524.

[192] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," in *2021 IEEE Symposium on Security and Privacy (SP).* IEEE, 2021, pp. 141–159.

[193] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, H. Yin, and Q. V. H. Nguyen, "A survey of machine unlearning," *arXiv preprint arXiv:2209.02299*, 2022.

[194] R. D. Cook and S. Weisberg, *Residuals and influence in regression.* New York: Chapman and Hall, 1982.

[195] G. Wu, M. Hashemi, and C. Srinivasa, "Puma: Performance unchanged model augmentation for training data removal," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8675–8682.

[196] P. W. Koh and P. Liang, “Understanding black-box predictions via influence functions,” in *International conference on machine learning.* PMLR, 2017, pp. 1885–1894.

[197] C. Wang, Y. Zhou, Q. Wen, and Y. Wang, “Improving load forecasting performance via sample reweighting,” *IEEE Transactions on Smart Grid*, vol. 14, no. 4, pp. 3317–3320, 2023.

[198] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[199] B. A. Pearlmutter, “Fast exact multiplication by the hessian,” *Neural computation*, vol. 6, no. 1, pp. 147–160, 1994.

[200] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, “Mlp-mixer: An all-mlp architecture for vision,” *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.

[201] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, “Gradient surgery for multi-task learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, 2020.

[202] A. Ben-Israel and T. N. Greville, *Generalized inverses: theory and applications.* Springer Science & Business Media, 2003, vol. 15.

[203] J. H. Gallier. (2010) Notes on the schur complement. [Online]. Available: https://repository.upenn.edu/cgi/viewcontent.cgi?article=1637&context=cis_papers

[204] C. Liu and I. M. Jaimoukha, “The computation of full-complexity polytopic robust control invariant sets,” in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 6233–6238.

[205] J. Fortuny-Amat and B. McCarl, “A representation and economic interpretation of a two-level programming problem,” *Journal of the operational Research Society*, vol. 32, pp. 783–792, 1981.

[206] S. J. Kazempour, A. J. Conejo, and C. Ruiz, “Strategic generation investment using a complementarity approach,” *IEEE transactions on power systems*, vol. 26, no. 2, pp. 940–948, 2010.

[207] N. Jorge and J. W. Stephen, *Numerical optimization.* Spinger, 2006.