Decision-focused Electricity Prices Prediction Framework for Demand Response Programs

Linwei Sang

***Abstract*—The abstract goes here.**

***Index Terms*—Price prediction, optimization model, model and data driven**

I. Introduction

WITH the growth of battery energy storage system (BESS) in power system, electricity prices prediction becomes important in BESS management and scheduling [1]. Electricity prices are decided by electricity supply-demand and become volatile due to the high penetration of renewables and deregulation of electricity market [2]. Hence, its accurate prediction is difficult. BESS is taken as a price-taker in the energy market, which schedules its charging/discharging time period based on predicted prices to gain arbitrage from the market[3]. For BESS optimal decisions, predicted prices from prediction model is transmitted to downstream BESS scheduling model, which makes the optimal charging/discharging decisions over the period of interest to maximize the benefits of BESS's arbitrage. This process is called "predict, then optimize" framework [7]. In this framework, various electricity price prediction methods have been proposed in the past five years.

However, more accurate electricity price prediction is not necessarily equivalent to more benefits of BESS's arbitrage. The main reason is that the objective of common prediction model is usually to minimize the prediction error, which describes the distance between predicted prices and true prices. In contrast, BESS focuses on maximizing benefits from the electricity price's fluctuation. This can be equivalent to minimizing the distance between actual optimal decisions under the predicted prices and oracle optimal decisions under the true prices, which is called the decision error in this paper. The decision error does not coincide with the prediction error.

This motivates us to further take the decision error from downstream BESS scheduling model

A. Related works

B. Background

Key: build the gap for the paper: the gap is ignoring the decision errors from the downstream model.

1) Price-taker demand response:

2) Electricity prices prediction:

3) "Prediction, then Optimize" scheme:

4) Prediction under the lens of optimization model:

5) Different types of storage:

Work in progress

C. Motivation

Conventional power system scheduling and optimization follow the "Predict, then Optimize" framework, where the connection between prediction model and optimization model is uni-direct. The prediction methods of current researches seldom consider the decision errors in training process. However, the minimizing of prediction errors is not equivalent to minimizing decision errors. So it is essential to consider both the prediction errors and decision errors in training the parameters of prediction model.

Inspired by the data-driven optimization model, this paper proposes a decision-focused electricity prices prediction framework by utilizing the decision errors from the downstream optimization model. This paper first formulates the price-taker demand response programs and electricity prediction models, utilizes mean-square-errors (MSE) and the regret to measure the prediction errors from prediction model and corresponding decision errors from optimization model, and derives the gradients of surrogate regret with respect to predicted prices. Based on the gradients of MSE and surrogate regret, this paper proposes a hybrid stochastic gradient decent (SGD) method for PDR decision. Finally, this paper discusses the decision errors reducing performance in different capacities of prediction model, which is compared with single MSE and single surrogate regrets.

D. Contributions

The main contributions of this work can be summarized as:

1) To the authors' best knowledge, this paper proposes a decision-focused electricity price prediction framework for price-taker demand response programs, which **first** takes the advantage of both prediction errors and decision errors to train the parameters of prediction models.
2) The regret is utilized to measure decision errors sourcing from prediction errors, and then the surrogate regret is further proposed to derive the gradients with respect to predicted prices for learning.
3) The hybrid loss function is proposed to balance prediction errors and decision errors, and its gradients are derived. In actual implementation, gradients of hybrid loss function are back-propagated to parameters of prediction model by twice back-propagation, *i.e.*, surrogate regret gradients and weighted MSE propagation.
4) Numerical experiments verify the proposed hybrid gradients can further reduce decision loss when MSE loss have reached its limits.

E. Related researches

Possible related researches:

1) Electricity prices prediction:

2) Machine learning and optimization: Introducing some machine learning techniques and the state-of-the-art research papers.

3) Decision-focused predicting methods: This paper reverses the calculation flow from downstream optimization model to upstream prediction model.

4) Utilizing the models from the heuristics:

5) Under decision-focused machine learning: Frontier researches have pay more attention to bridging the gap between machine learning methods with conventional optimization methods. The machine learning method provides more data insight to optimization model, while in the same time the optimization model leads to the more interpretability of complex machine learning models. (Possible networks: input convex neural networks, cvxlayers) In the area of smart grid, the connection can have concrete application scenarios.

F. Organization of this paper

The rest of this paper is organized as follows. Section II presents the formulation of conventional prediction and then optimization framework. Section III proposes decision-focused prediction framework and hybrid stochastic gradient methods to training prediction model. Section IV evaluates the effectiveness of proposed framework and method by numerical experiments. Section V summarizes this paper and envisions possible future direction.

II. Problem Formulation

Overall formulation (price taker). Overall illustration on the overall framework.

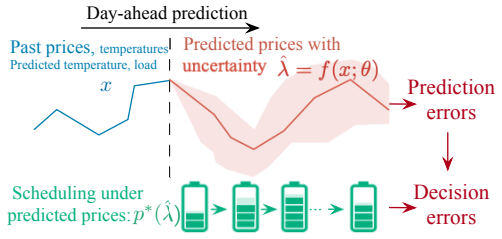


Fig. 1. Scheduling under prediction framework.

A. Price-based demand response programs

Refer to Ref. [1]

1) Battery storage formulation: The battery energy storage system (BESS) usually works at charging state when electricity price is low and at discharging state when electricity price is high. The operation of storage system usually follows the electricity price signals to maximize the benefits from markets as:

$$\max_{p} c(p;\lambda) = \lambda^T p = \sum_{t=1}^{T} \lambda_t p_t \tag{1}$$

where $p \in \Phi_{BESS}$

where its objective is to maximize the net energy benefits by optimizing its operating power p (MW) under market electricity prices λ ($/MWh), denoted by $c(p;\lambda)$; the feasible region (Φ_{BESS}) of decision variables p is subject to a set of operation and technical constraints as follows:

$$p = p_{dis} - p_{ch} \tag{2}$$

Equation (2) illustrates the operating power p of BESS is composed of charging and discharging parts; p_{dis} denotes the charging of BESS from grid, and p_{ch} denotes the discharging of BESS to grid.

$$\begin{aligned} &E(t) = E(t-1) + \eta_{ch} p_{ch}(t) - \frac{p_{dis}(t)}{\eta_{dis}} \quad t = 2,..,T \\ &E_{min} \le E(t) \le E_{max} \end{aligned} \tag{3}$$

Equation (3) ensures the stored energy $E(t)$ in the BESS at time t lies in allowable range;, E_{min}, E_{max} refer to the minimum, maximum capacity of the battery system.

$$\begin{aligned} &0 \le p_{dis} \le P_{dis}^{max} \\ &0 \le p_{ch} \le P_{ch}^{max} \\ &0 \le p_{dis} \le M\mu_{dis} \\ &0 \le p_{ch} \le M\mu_{ch} \\ &\mu_{dis}(t) + \mu_{ch}(t) \le 1 \quad t = 1,...,T \end{aligned} \tag{4}$$

Equation (4) prevents the simultaneously charging and discharging of BESS by utilizing the big-M method; M is the large positive number; μ_{dis} and μ_{ch} are binary indicators of discharging and charging state, where 1 means in the state and 0 means the opposite; P_{dis}^{max} and P_{ch}^{max} are the maximum values of charging and discharging states.

$$\sum_{t=1}^{T} \mu_{dis}(t) + \mu_{ch}(t) \le N_{cycle} \tag{5}$$

Equation (5) limits the number of full cycles per day (N_{cycle}).

2) Price taker distributed generators:

3) Shiftable load: Shiftable load can schedule their energy demand from peak hours to off-peak hours when prices are low in the scheduling horizon. The operation of shiftable load can follow the day-ahead prices information to minimize its cost as:

$$\max_{p} c(p;\lambda) = -\lambda^T p = \sum_{t=1}^{T} -\lambda_t p_t \tag{6}$$

where $p \in \Phi_{SL}$

The feasible region Φ_{SL} of shiftable load is decided by devices' operation interval $[T_{n,start}, T_{n,end}]$ and their required operation time in Equation (7).

$$\begin{aligned} &p_t = \mathbb{I}_{n,h} p_{n,t} \\ &T_{n,start} \le T_{n,\omega} \le |T_{n,end} - T_{n,last}| \\ &T_{n,last} \le T_{n,end} - T_{n,start} \end{aligned} \tag{7}$$

4) Price-based demand response programs: The general form of price-based demand response programs is formulated as a linear objective model as follows:

$$\max_{p} c(p;\lambda) = \lambda^T p = \sum_{t=1}^{T} \lambda_t p_t \tag{8}$$

where $p \in \Phi$

B. Electricity price prediction model

Above day-ahead price-based scheduling model is based on the prediction of day-ahead electricity prices. This part focus on the price prediction model formulation. Its general form is as follows:

$$\hat{\lambda} = f(x) \tag{9}$$

where $\hat{\lambda}$ is the prediction electricity prices and x the input features; $f(\cdot)$ denotes the prediction model. From input raw datasets to output predicted prices, the whole process is composed of four procedures: i) data pre-processing, ii) feature engineering, iii) model selection, iv) training procedures, v) prediction target.

1) Data pre-processing: Data pre-processing transforms raw datasets into model's input and output datasets, including filling missing values, clearing outliers, and normalizing the dataset.

2) Feature engineering: Feature engineering selects and formulates the feature vectors from the processed dataset for inputting the prediction model. The electricity prices is usually influence by historical load and prices, future load and temperature, and some calendar factors including weekday/weekend, holiday effects, and day of the year. For more features input, the squares of future load and temperature are formulated and added into the feature vector.

3) Model selection: Prediction problem is formulated as a regression problem, which maps the input feature vectors into the output continuous prediction values. A lot of regression models are applied in the smart grid for electricity prices, from conventional linear regression in statistical area to burgeoning neural networks in deep learning area. These models feature various strengthens and data input.

In this work, we focus on comparing two kinds of model: i) the linear regression model in Equation (10); ii) the ResNet model in Equation (11).

$$\hat{\lambda} = f(x;\theta) = \theta_x^{lr} x + \theta_b^{lr} \tag{10}$$

$$\begin{aligned} y_{l+1} &= \sigma(\theta_y^{nn} y_l + \theta_x^{nn} x_l + \theta_b^{nn}) \quad l = 1, ..., N \\ \hat{\lambda} &= y_N \end{aligned} \tag{11}$$

4) Training procedures: Training prediction model for better generalization follows these basic procedures: i) A subset of full processed data is selected as testing dataset randomly; ii) the remaining is further split into training and validation set randomly; iii) then we train the prediction model on the training set, evaluate its accuracy on validation set, and tune models' hyperparameters for high accuracy; iv) finally we test the trained model on the testing set for model evaluation and comparison.

5) Prediction target: The target of training prediction model is to minimize the distance between predicted prices and actual prices. The distance is usually described by loss function as shown in definition 1.

Definition 1 (MSE loss of Prediction)**.** *The mean square error (MSE) of prediction between the predicted prices and actual prices is defined as:*

Algorithm 1 Stochastic gradient decent algorithm for prediction

1: **Input:** Raw dataset, prediction model; hyperparameter: batch size N, learning rate α;
2: **Data processing:** Data pre-processing, featuring engineering, and train-validation-test dataset dividing;
3: **while** Not converge **do**
4: Sampling N data point;
5: **for** i = 1, ... N **do**
6: Predict prices: $\hat{\lambda} = f(x)$
7: Calculate gradients of MSE according to Equation (14):

$$\frac{\partial L^{MSE}(\hat{\lambda})}{\partial \hat{\lambda}} \leftarrow \frac{1}{T}(\hat{\lambda}_i - \lambda_i)$$

8: Accumulate gradients of MSE:

$$\frac{\partial L^{MSE}}{\partial \hat{\lambda}} \leftarrow \frac{\partial L^{MSE}(\hat{\lambda})}{\partial \hat{\lambda}} + \frac{\partial L_i^{MSE}(\hat{\lambda})}{\partial \hat{\lambda}}$$

9: **end for**
10: Update model parameters:

$$\theta \leftarrow \theta - \alpha \frac{\partial L^{MSE}}{\partial \hat{\lambda}} \frac{\hat{\lambda}}{\theta}; \frac{\partial L^{MSE}}{\partial \hat{\lambda}} \leftarrow 0$$

11: Evaluate the model in validation dataset;
12: **end while**
13: **Output:** Final prediction model.

$$L^{MSE}(\hat{\lambda}, \lambda) = \frac{1}{T} \frac{||\hat{\lambda} - \lambda||_2^2}{2} = \frac{1}{T} \sum_{t=1}^{T} \frac{(\hat{\lambda}_t - \lambda_t)^2}{2} \tag{12}$$

Then the target of prediction model is formulated in Equation (13).

$$\begin{aligned} &\min_{\theta} L^{MSE}(\hat{\lambda}, \lambda) \\ &\text{s.t. } \hat{\lambda} = f(x;\theta) \end{aligned} \tag{13}$$

Lemma 1 (Gradients of MSE to predicted prices)**.** *The mean square error (MSE) of prediction between the predicted prices and actual prices is defined as:*

$$\begin{aligned} &\frac{\partial L^{MSE}(\hat{\lambda}, \lambda)}{\partial \hat{\lambda}} = \frac{1}{T} \sum_{t=1}^{T} (\hat{\lambda}_t - \lambda_t) \\ &\text{s.t. } \hat{\lambda} = f(x;\theta) \end{aligned} \tag{14}$$

C. Back-propagation preliminary

Explain the back-propagation procedures preliminaries

D. Bridging prediction and optimization

Above formulation separates the predicting stage with the optimizing stage, and the connection between two stages is uni-direct where the upstream prediction model delivers its prices to downstream optimization model explicitly. In the same

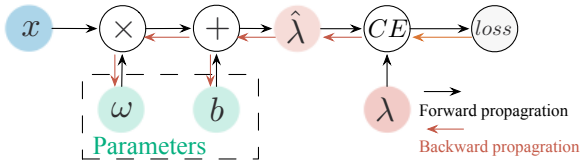


Fig. 2. The illustration of calculation graph by Pytorch.

time, the prediction errors are also delivered to the downstream implicitly, which lead to the decision errors. Current works [TO DO: Ref.] focus on utilizing these prediction errors to train and evaluate the prediction model, while few considers downstream decision errors from upstream prediction errors.

This work focuses on utilizing downstream decision errors to improve the prediction model. The prediction model considers not only the prediction errors but the decision errors. As shown in Fig. 3, the downstream optimization model receives the predicted prices, calculates the decision errors, and back-propagates the decision errors to the prediction model. The concrete error calculation and back-propagation process are detailed in the following section.

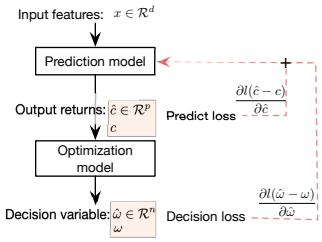


Fig. 3. The transforming of uni-direct to bi-direct (Replacing this figures).

III. Methodology

TO DO: Overall framework in first

A. General framework

The major drawback of above "Predict, then Optimize" uni-direct framework is that it does not delivery of prediction errors to optimization model. So this section first measures the decision loss from prediction errors in Equation (8) by the *regret*, then derives derivatives of the regret to predicted prices, and finally back-propagates the gradients to update the parameters of prediction model as shown in Fig. 4.

[TO DO: Consider generalizing the framework in formula]

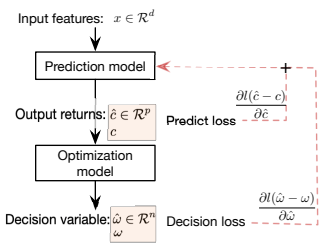


Fig. 4. Overall framework (Replacing this figures).

B. Regret of decision

In this paper, *regret* describes distance between the cost of decision under predicted values and that under actual values TO DO: Ref., which is widely utilized to measure the performance of online learning algorithms, e.g., multi-armed bandits, reinforcement learning, Thompson sampling.

1) Regret loss of PDR decision: In the proposed electricity prices prediction for PDR, we measure the difference of the predicted decisions under the predicted prices and the optimal decisions under actual prices in definition 2. Low regret loss means asymptotically optimal decisions.

Definition 2 (Regret of PDR decision)**.** *The regret of PDR decision is defined as the gap between the benefits under the predicted decisions and that under optimal decisions:*

$$\begin{aligned} regret(\hat{\lambda}, \lambda) =& \lambda^T P^*(\lambda) - \lambda^T P^*(\hat{\lambda}) \\ \text{where } P^*(\lambda) =& P^*_{dis}(\lambda) - P^*_{ch}(\lambda) \end{aligned} \tag{15}$$

where $P^*(\lambda)$, $P^*(\hat{\lambda})$ denote the predicted decisions and the optimal decisions; λ denotes the actual electricity prices in the market.

2) Discussion: Due to the MILP formulation of PDR, its feasible region can be described as a collection of polyhedrons, so its optimal decision probably lies in the extreme points of some polyhedron. Taking two-dimension polyhedrons as an example, Fig. 5 shows two different predicted electricity prices with the same predicted errors may lead to different decisions and corresponding different *regrets*. The decision $P(\lambda)$ under λ is same as $P(\hat{\lambda}_1)$ under $\hat{\lambda}_1$, while different from $P(\hat{\lambda}_2)$ under $\hat{\lambda}_2$. This also applies to *regret* of different predicted prices.

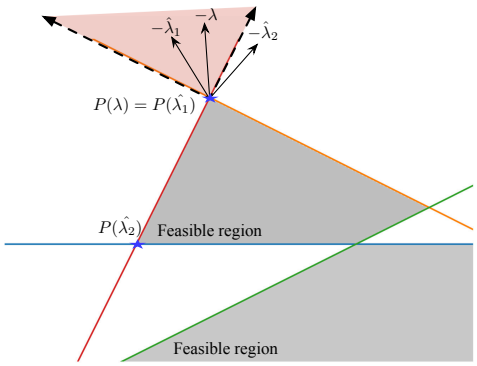


Fig. 5. Geometric illustration of different predicted electricity prices with same predicted error but different *regret* losses.

Above example illustrates the *regret* loss of prediction focuses on the impact of prediction errors on decisions. The prediction model should utilize *regret* losses to update its parameters, but the gradients of *regret* loss to predicted prices is hard to calculate directly.

C. Tractable transformation of regret

However, the *regret* with respect to predicted prices $\hat{\lambda}$ is discontinuous and non-tractable. Based on Ref. [7], we

formulate the tractable loss function L^{regret} of the *regret* by combining MSE loss.

$$\begin{aligned} regret(\hat{\lambda}, \lambda) &= \lambda P^*(\lambda) - \alpha\hat{\lambda}P^*(\hat{\lambda}) + [\alpha\hat{\lambda}P^*(\hat{\lambda}) - \lambda P^*(\hat{\lambda})] \\ &= [\alpha\hat{\lambda}P^*(\hat{\lambda}) - \lambda P^*(\hat{\lambda})] + c^*(\lambda) - \alpha c^*(\hat{\lambda}) \end{aligned} \tag{16}$$

$$regret(\hat{\lambda}, \lambda) \leq \inf_{\alpha} \left\{ \max_{P\in\Phi}\{\alpha\hat{\lambda}P - \lambda P)\} - \alpha c^*(\hat{\lambda}) \right\} + c^*(\lambda) \tag{17}$$

Inequation (17) is actually a dual form of an equation, and the optimal value α of left-hand formula tends to ∞. As α is getting larger, the term of λP tends to be negligible and the optimal P of inner maximization problem tends to be $P(\hat{\lambda})$, which recovers Equation (16). So the dual form of Equation (16) is established.

$$\begin{aligned} regret(\hat{\lambda}, \lambda) = \lim_{\alpha\to\infty} &\left\{ \max_{P\in\Phi}\{\lambda^T P + \alpha\hat{\lambda}^T P\} - \alpha c^*(\hat{\lambda}) \right\} \\ &+ c^*(\lambda) \end{aligned} \tag{18}$$

When hybrid with previous prediction model formulation (9), the loss reducing target function of the prediction model can be further extended in Equation (19). The first equality of above derives from the fact $P^*(\alpha_i f(x_i)) = P^*(f(x_i))$ for any $\alpha_i \geq 0$, which is proved in appendix. The second equality derives from the intuition that as all α_i tends to be ∞, they tend to be the same and can be replaced by a single variable α. The first inequality derives from setting α as 2 to get an estimate of upper bound in particular. And the second inequality relaxes optimal value of $P^*(2f(x_i))$ under $c(\hat{\lambda})$ by a feasible decision value $P^*(\lambda)$. The detailed illustrations are attached in appendix.

Definition 3 (Surrogate regret loss of PDR decision)**.** *Given predicted prices $\hat{\lambda}$ and actual prices λ, the surrogate regret loss of PDR decision is defined as follows:*

$$\begin{aligned} L^{regret}(\hat{\lambda}, \lambda) &= \max_{P\in\Phi}\{\lambda^T P - 2\hat{\lambda}^T P\} - 2\hat{\lambda}P^*(\lambda) + c^*(\lambda) \\ &= (\lambda - 2\hat{\lambda})^T P^*(\lambda - 2\hat{\lambda}) - 2\hat{\lambda}P^*(\lambda) + c^*(\lambda) \end{aligned} \tag{20}$$

Remark 1 (Properties of surrogate *regret* loss)**.** *Given predicted prices and actual prices, proposed surrogate regret loss holds the following properties:*

1. $regret(\hat{\lambda}, \lambda) \leq L^{regret}(\hat{\lambda}, \lambda)$;

2. $L^{regret}(\hat{\lambda}, \lambda)$ is a concave function of predicted electricity prices $\hat{\lambda}$.

Lemma 2 (Gradients of the surrogate *regret* loss to predicted prices)**.** *Based on the definition of surrogate regret of PDR decision, the gradient of regret to prediction price is derived for later model training:*

$$\frac{\partial L^{regret}(\hat{\lambda}, \lambda)}{\partial\hat{\lambda}} = -2(P^*(\lambda) + P^*(\lambda - 2\hat{\lambda})) \tag{21}$$

Proof. The proof of Equation (21) is as follows [TO do: Refer to some perturbation optimization analysis]: □

$$\begin{aligned} \frac{\partial L^{regret}}{\partial\hat{\lambda}} &= \frac{\partial \max_{P\in\Phi}\{\lambda^T P - 2\hat{\lambda}^T P\} - 2\hat{\lambda}P^*(\lambda) + c^*(\lambda)}{\partial\hat{\lambda}} \\ &= \frac{\partial \max_{P\in\Phi}\{\lambda^T P - 2\hat{\lambda}^T P\}}{\partial\hat{\lambda}} - 2P^*(\lambda) \\ &= -2P^*(\lambda - 2\hat{\lambda}) - 2P^*(\lambda) \end{aligned}$$

D. Hybrid SGD method for PDR decision

MSE and surrogate regret focus on the prediction errors and decision errors individually, which views prediction loss from the perspectives of prediction model and decision model. But they can be combined in training for better trade-off, and hence this paper proposes a decision-focused stochastic gradient decent (SGD) method for training prediction model by utilizing the hybrid loss function (22) and its derivatives (23).

Definition 4 (Hybrid loss function)**.** *The hybrid loss is the weighted sum of MSE (12) and surrogate regret (20).*

$$\begin{aligned} L^{comb} &= L^{regret} + \epsilon L^{MSE} \\ &= (\lambda - 2\hat{\lambda})^T P^*(\lambda - 2\hat{\lambda}) - 2\hat{\lambda}P^*(\lambda) + c^*(\lambda) \\ &+ \epsilon\frac{1}{T}\frac{||\hat{\lambda} - \lambda||_2^2}{2} \end{aligned} \tag{22}$$

where ϵ is a weighted coefficient on MSE, which implies the emphasis extent of prediction errors.

Lemma 3 (Hybrid gradients of the combining *regret* and MSE to predicted prices)**.** *The hybrid gradients of combining regret and MSE to predicted prices are derived from Equation (14) and (21):*

$$\frac{\partial L^{comb}}{\partial\hat{\lambda}} = \frac{\partial L^{regret}}{\partial\hat{\lambda}} + \epsilon\frac{\partial L^{MSE}}{\partial\hat{\lambda}} \tag{23}$$

The gradients of hybrid loss function can be calculated explicitly theoretically by 23. However, in actual implementation, the gradients of MSE are calculated implicitly with the help of the Autograd tool [11] in Pytorch, and in contrast the gradients of surrogate regret are calculated explicitly by solving MILP problem in 21.

To tackle with these two distinct ways of gradients calculation, the updating of hybrid gradients is divided into three steps: 1) calculate the gradients of surrogate regret based on Equation (21), then feed the gradients into the tensors of predicted prices, back-propagate the gradients for leaf nodes in calculation graph, and retain the gradients for leaf nodes; 2) calculate and back-propagate the weighted MSE loss of prediction prices to the same leaf nodes; 3) update the values of leaf nodes based on accumulated gradients. The hybrid gradients updating is actually achieved by twice back-propagating and only once updating.

Based on the above derivation and discussion, this paper proposes the hybrid SGD method for PDR decision to achieve the trade-off between the prediction errors and decision errors. It should be noted that proposed method can apply for both

$$
\begin{aligned}
&\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \lim_{\alpha_i \to \infty} \left\{ \max_{P\in\Phi}\{\lambda^T P - \alpha_i f(x_i)^T P\} - \alpha_i c^*(\hat{\lambda}) \right\} + c^*(\lambda) \\
=&\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \lim_{\alpha_i \to \infty} \left\{ \max_{P\in\Phi}\{\lambda^T P - \alpha_i f(x_i)^T P\} - \alpha_i f(x_i)^T P^*(\alpha f(x_i)) + c^*(\lambda) \right\} \\
=&\min_{\theta} \lim_{\alpha \to \infty} \frac{1}{n}\sum_{i=1}^{n} \left\{ \max_{P\in\Phi}\{\lambda^T P - \alpha f(x_i)^T P\} - \alpha f(x_i)^T P^*(\alpha f(x)) + c^*(\lambda) \right\} \\
\leq&\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \max_{P\in\Phi} \left\{ \lambda^T P - 2f(x_i)^T P \right\} - 2f(x_i)^T P^*(2f(x_i)) + c^*(\lambda) \\
\leq&\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \max_{P\in\Phi} \left\{ \lambda^T P - 2f(x_i)^T P \right\} - 2f(x_i)^T P^*(\lambda) + c^*(\lambda)
\end{aligned}
\tag{19}
$$

Algorithm 2 Decision-focused SGA method for PDR decision

1: **Input:** Raw dataset, prediction model; hyperparameter: batch size N, learning rate α;
2: **Data processing:** Data pre-processing, featuring engineering, and train-validation-test dataset dividing;
3: **Training:**
4: **while** Not converge **do**
5: Sampling N data point;
6: **for** i = 1, ... N **do**
7: Predict prices: $\hat{\lambda} = f(x)$;
8: **Calculate** the weighted gradients of MSE based on the Autograd tools;
9: **1st Back-propagate** the weighted gradients of MSE to the parameters of prediction model;

$$\frac{\partial L^{Comb}}{\partial \hat{\theta}} += \frac{\partial L^{MSE}}{\partial \hat{\theta}}$$

10: **Calculate** gradients of surrogate *regret* according to Equation (15):

$$\frac{\partial L^{regret}(\hat{\lambda})}{\partial \hat{\lambda}} = \lambda^T P^*(\lambda) - \lambda^T P^*(\hat{\lambda})$$

11: **2nd Back-propagate** the gradients of surrogate *regret* to the parameters of prediction model;

$$\frac{\partial L^{Comb}}{\partial \theta} += \frac{\partial L^{regret}}{\partial \hat{\lambda}} \frac{\partial \hat{\lambda}}{\partial \theta}$$

12: **end for**
13: Update model parameters in batches:

$$\theta = \theta - \alpha \frac{\partial L^{Comb}}{\partial \theta}; \frac{\partial L^{Comb}}{\partial \theta} = 0$$

14: Evaluate the model in validation dataset;
15: **end while**
16: **Output:** Final prediction model.

simple linear prediction models and complex deep learning models.

As shown in algorithm 2, firstly, the data processing conducts pre-processing, feature engineering, and dataset partition as mentioned in section II-B; secondly, training processing samples mini-batch datasets from the whole dataset, calculates the gradients of MSE and surrogate *regret*, accumulates each points' gradients in one mini-batch, and updates the parameters of prediction model in batches. The way of mini-batch updating reduces the updating gradients' number for acceleration. After each batch updating, the model in validation dataset is evaluated. In actual implementation with Pytorch package featuring autograd tools, the hybrid loss back-propagating process is divided into three processes: 1) calculate surrogate *regret* gradients, back-propagates newly gradients to model parameters, and retain the gradients; 2) calculate weighted MSE gradients, back-propagates the gradients again, and accumulate the above two gradients with respect to corresponding parameters. Finally, the trained model is applied for online electricity prices prediction. Rewrite this part

E. Discussion

IV. Experiments

To verify the effectiveness of proposed gradients and methods, this paper utilizes six-year hourly real electricity prices and related temperature, load information from PJM for numerical experiments[12]. All the proposed methods are implemented by python with Pytorch framework for prediction models and Cvxpy framework for optimization models, which are trained on a MacBook Pro laptop with RAM 16 GB, CPU Intel Core I7 (2.6GHz).

A. Datasets and models

This work first pre-processes the raw electricity-related data to form the prediction dataset, then extracts the input features and output prices, trains different day-ahead prediction models, and finally evaluates different models and loss function decision by multi-metrics.

1) Data preparing: We construct the input features from three aspects: 1) the past day information: the hourly load, temperature, and temperature's square in the past day; 2) the prediction day information: the prediction temperature and its square in the prediction day; 3) calendar effects: the indicators of weekday, holiday, and day of the year as shown in Table I.

Then all the input features are standardized considering mean and standard variance to formulate the final feature vectors.

As the electricity prices fluctuate a lot, the output of the prediction model is set to be the log of electricity prices, which is assumed to follow log-normal distribution [5]. After feature

TABLE I
THE INPUT FEATURES OF PRICE PREDICTION MODELS.

Features' type	Components
Past day	load, temp, $(\text{temp})^2$
Prediction day	temp, $(\text{temp})^2$
Calendar effects	$\mathbb{I}(weekday)$, $\mathbb{I}(holiday)$, $\sin(2\pi \times DOY)$, $\sin(2\pi \times DOY)$

It should be noted that temp is the abbreviation of temperature, $\mathbb{I}(\cdot)$ refers to the indicator function, DOY is the abbreviation of Day of the Year.

engineering, the 20% of all above pre-processed datasets are split randomly into test set for model evaluation; the 20% of the rest are further split randomly into validation set for training early stop, and the final rest is the training set for training the parameters of the model.

2) Model initializing: Fig. 6 visualizes the whole structure of the prediction process, whose key lie in the design of ResNet model which maps constructed input features to output log of prices. As shown in Fig. 6, ResNet model is composed of full connected (FC) layers and residual connected (RC) layers: the parameters of FC layers is initialized randomly, and whereas the parameters of RC layers is initialized by linear least squares $(X^TX)^{-1}X^TY$ [13].

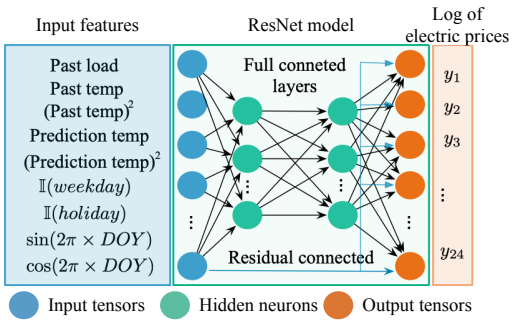


Fig. 6. The structure of ResNet model.

For PDR decision model, we normalize the energy capacity into 1, and the charging/discharging power depends on the daily depths of charging/discharging. So the unit of regret in our case is \$/MWh.

3) Hyper-parameter setting: The hyperparameter setting also has two parts, *i.e.*, ResNet model and PDR model's hyper-parameters as shown in Table II.

TABLE II
THE INPUT FEATURES OF PRICE PREDICTION MODELS.

ResNet model		**PDR model**	
Hyper-parameters	Values	Hyper-parameters	Values
Optimizer	Adam	Hyper-parameters	Values
Learning rate	1e-6	Depth of charging	0.5
Hidden layers	[50, 50]	Depth of discharging	0.5
Batch size	100	η_{ch}	0.90
Dropout	0.2	η_{dis}	0.92

TABLE III
COMPARISON OF DIFFERENT PREDICTION MODELS.

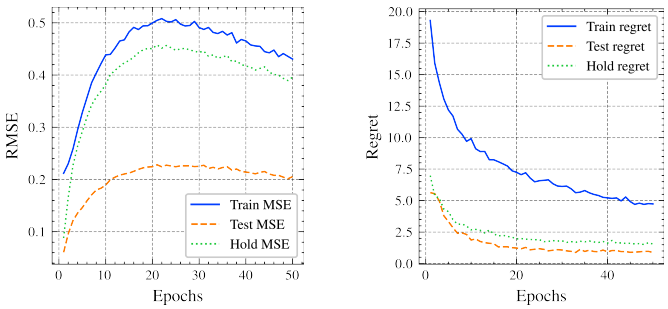
Errors metrics	RMSE	MAPE	Regret
Decision-focused prediction	0.199	0.0466	1.343
MSE-based prediction	0.320	0.078	10.47
MLP	0.0272	0.0399	2.208
Random forest	0.0267	0.0415	1.433

It should be noted that the hidden layers of MLP is set as 300; and the depth of random forest method is set as 30.

B. Performance of the proposed prediction model

1) Training process: Fig. 7 visualizes MSE and regret changing in training the decision-focused ResNet prediction model under the setting of Table II. As training epochs increases, the MSE of three split datasets increase firstly and then decrease at around 23rd epoch in Fig. 7(a). In contrast, the regrets of three split datasets decrease shapely at first and then flat at around 45th epoch in Fig. 7(b).

The different loss changing curves of MSE and regret show that predicted prices under initialized parameters of ResNet have low MSE but high regret. This implies MSE does not correspond to regret in some circumstance. The mere consideration of MSE in training prediction model is not comprehensive.



(a) The changing process of MSE (b) The changing process of regret

Fig. 7. The errors changing in the training process.

2) Models' comparison: To verify the effectiveness of proposed model, we compare the proposed decision-focused prediction (DFP) model with only MSE-based models, multi-layer perceptron (MLP), and random forest models, which are widely applied in electricity prices prediction. The measuring metrics of prediction models' performance in test set mainly have two components: prediction accuracy metrics, *i.e.*, root-mean-square-error (RMSE), mean-absolute-percentage-error (MAPE) and decision accuracy, *i.e.*, regret of PDR.

The results of different prediction models are compared in Table III. It is obvious that proposed DFP model achieves lower decision regret than other methods, though endures high MSE and MAPE compared to MLP and random forest method. Compared with MLP methods, DFP reduces 34% regret and further increases 0.865 dollars benefits per MWh on average; compared with random forest model, DFP reduces 6% regret and further increases 0.090 dollars benefits per MWh on average.

C. Analysis of the proposed prediction framework

The proposed decision-focused prediction framework and SGA training algorithm can apply to not only complex but the simple prediction model by changing network structures. To unfold the impact of hybrid loss, this part utilizes the loss to train the linear model (10) and ResNet model (11) respectively. In each prediction model, we further analyze the effect of different weight parameters ϵ in hybrid loss for deciding its proper value.

1) Linear prediction model: The proposed hybrid loss is first utilized to train the simple linear prediction models (10) with different ϵ. For comparison, the surrogate regret loss means ϵ takes the value of 0, whereas MSE loss means ϵ takes a large value to ignore the effects of surrogate regret part in hybrid loss.

Performance evaluation: After 100 times' training, we evaluate the trained model by the same metrics in previous part, *i.e.*, RMSE, MAPE, and regret of PDR.

TABLE IV
LOSS FUNCTION COMPARISON OF LINEAR PREDICTION MODEL.

Loss	ϵ	RMSE	MAPE	Regret
MSE	/	3.395	1.03	18.780
Surrogate regret	/	3.239	0.985	16.214
Hybrid loss	25	3.229	0.988	7.018
Hybrid loss	50	2.558	0.783	3.133
Hybrid loss	100	1.586	0.482	2.154
Hybrid loss	200	0.648	0.19	2.425

As shown in Table IV, with the increasing of ϵ from 25 to 200, the RMSE and MAPE is decreasing correspondingly. But the regret of PDR is decreasing at first and then increasing when ϵ is 200.

MSE and surrogate regret loss feature high RMSE, MAPE, and regret when utilized alone. Obviously, the proposed hybrid loss can help guide the small-capacity prediction model to predict and make decisions more accurately, which is in essence the combination of above two losses.

Errors in different time intervals: When ϵ is 25, the RMSE and MAPE of MSE, surrogate regret, hybrid loss are similar, but hybrid loss achieves lower regret compared the others. Then we further investigate the hourly prediction errors each time period in Fig. 8.

As shown in Fig. 8, though the whole prediction errors under above three losses are similar, the time distribution of prediction errors under MSE and surrogate regret embodies high variation than that under hybrid loss. This verifies the effectiveness of hybrid loss for guiding the prediction model to make the prediction for less decision errors.

Results analysis: Above numerical experiments provide valuable insights of the proposed hybrid loss in small capacity models: i) hybrid loss can take advantage of the gradients' information from both prediction errors and decision errors to achieve more accurate prediction and decision; ii) With the increasing emphasis on MSE, the prediction errors is reducing all alone while decision errors go down first and up after.

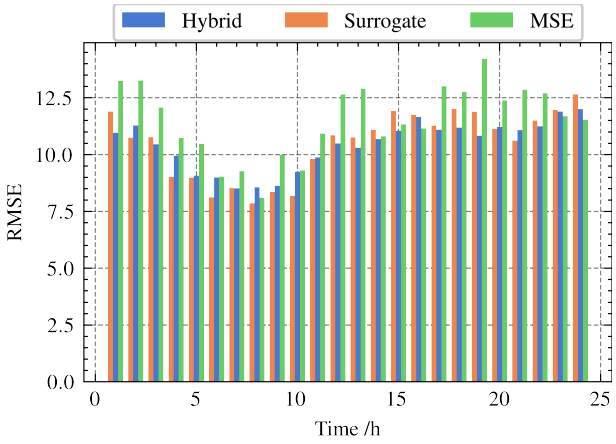


Fig. 8. RMSE in different time interval of different loss function by linear model.

2) ResNet prediction model: Then, the hybrid loss is utilized to train the complex ResNet models (11) with different ϵ. Similarly, the surrogate regret and MSE loss are compared in this situation.

Performance evaluation: After 50 times' training, we evaluate the trained ResNet model by the same metrics previously. The prediction and decision performance of different loss function design is measure in Table V.

TABLE V
LOSS FUNCTION COMPARISON OF RESNET PREDICTION MODEL.

Loss	ϵ	RMSE	MAPE	Regret
MSE	/	0.320	0.078	10.47
Surrogate regret	/	0.578	0.129	0.899
Hybrid loss	25	0.320	0.085	1.048
Hybrid loss	50	0.199	0.0466	1.343
Hybrid loss	100	0.153	0.035	1.938
Hybrid loss	200	0.1439	0.03185	3.588

Similarly to linear models, with the increasing of ϵ from 25 to 100, prediction models' RMSE and MAPE are decreasing correspondingly, but models' regret are increasing, which means the models' prediction accuracy is improved at the cost of reducing decision accuracy.

Comparing with single loss design, MSE features low RMSE and MAPE but high regret; while surrogate regret features high RMSE and MAPE but low regret. Though hybrid loss models endure a little higher regret than surrogate regret model, it apparently reduces the RMSE and MAPE of single loss models.

Errors in different time intervals: When ϵ takes the value of 25, the RMSE of MSE and hybrid loss is similar, which is lower than surrogate regret model. We further investigate hourly prediction errors.

As shown in Fig. 9, though the hourly prediction errors of surrogate regret is higher than those of MSE and hybrid loss, it has the lowest decision errors. The time distribution of prediction errors under MSE embodies large variation with high prediction errors in the afternoon (12, 13, and 16), which leads to wrong decisions in PDR. In contrast, the time distribution

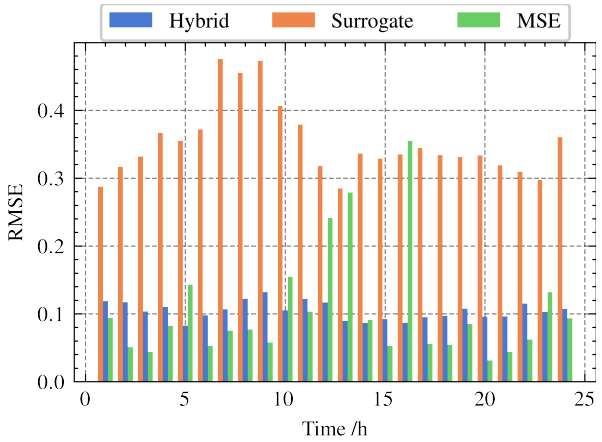


Fig. 9. RMSE in different time intervals of different loss function by ResNet model.

of prediction errors under hybrid loss embodies low variation and time-stable prediction loss.

Results analysis: Numerical experiments in ResNet model help further capture the insights of hybrid loss in large capacity prediction models: i) the updating directions from prediction errors and decision errors are not consistent, which needs to be traded-off; ii) the gradients from surrogate regret tends to flatten the time-distribution of prediction errors, which leads to the more accurate decision.

3) Discussion: Above discusses the impact of hybrid loss in small-capacity (linear model), large-capacity (ResNet) prediction models and analyzes the effect of weight parameter ϵ numerically in each prediction model, which can broaden its application in various models and scenarios. From numerical experiments, the prediction accuracy of prediction model is not consistent with its decision accuracy, that is, low prediction errors does not mean low decision errors. The proposed hybrid loss can take advantage of both prediction errors and decision errors to achieve more accurate prediction and decision by combining the gradients from MSE and surrogate regret.

Application analysis: Small-capacity and large-capacity models can be interpreted as following two typical cases:

- Case 1 : electricity prices is hard to predict accurately in some circumstances due to its uncertainty. In this case, hybrid loss can help reduce both prediction errors and decision errors;
- Case 2 : electricity prices also show regularity and can be predicted accurately in some circumstances. In this case, hybrid loss can also help reduce prediction errors at the cost of increasing some decision errors comparing to surrogate regret.

Weight parameters analysis: Consider plotting the MSE, MAPE, and regret with respect to the ϵ. Add figures here.

Gradients analysis: Roughly statement: it is hard to direct compare the value of two kind loss function regret and MSE, so to be consistent, we compare their gradients with respect to predicted prices.

Comparison

V. Conclusion

A. Summary

B. Future work

1. Extension to rolling prediction model, real-time prediction model, economic dispatch model...
2. Adaptive weighted parameters' formulation.

Appendix A
Equivalent form

The proof of $P^*(\alpha_i f(x_i)) = P^*(f(x_i))$ is as following:

Proof. □

Appendix B
Bayesian minimizer

Better illustrations.

Appendix C
ResNet stacked models

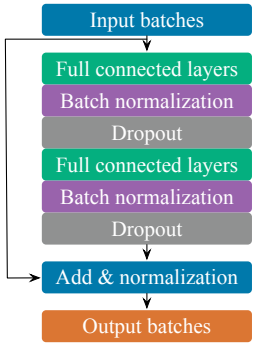


Fig. 10. ResNet stacked models

Appendix D
Experiments part

1) Results Analysis: Possible results analysis for the test dataset:

A. The impact of different weight parameters

To better illustrate the impact of weight parameters, we further analyze the changes of RMSE, MAPE, and regret of PDR decision with regard to ϵ in linear and ResNet models.

B. Gradients changing in the training process

1) Visualizing different kinds of gradients in SGD updating (if possible): The length, width of the picture denotes the training epochs and the gradients of different loss functions.

Acknowledge

The authors would like to thank ... for their valuable advice and instructions in completing this paper.

Notations

- Compare the work in the way of tables and figures;
- Consider visualizing different gradients of SPO and MSE;
- **Terminology consideration and discrimination:** the "predicted prices", "actual prices" and "true prices", which one is better ? Actual decisions and **oracle** decisions are both optimal decisions under different prices.
- Summarize the distinct description of loss function and gradients.
- From prediction errors, loss function, gradients of loss function, and error of decision, definition of decision errors, the flow of what we want to do !!! The logic line of this problem: The prediction errors lead to decision errors, then we utilize the MSE and regret metrics to measure these errors, and finally the utilization is based on the gradients' deduction of proposed methods.
- Principles: try to flow together, clear, concise, consistent and coherent.
- The word choice of 'forecasting' or 'predicting', and which is better? Prediction and forecast is more common in this framework.
- We should compare test curve or training curves ?
- The training may not be stable for some time, and prediction model may not be stable?
- The log choice of predicted prices ?
- Compare different predicted prices and decision curves under different situations;
- Comparable loss functions;
- The first pages to illustrate the main contribution of this framework, consider going into the main theme directly;
- The colors in bar-plot are not explicit to clear delivery;
- The clear delivery of the paper (who to emphasize and);
- Concise and explicit;
- Turn "combined" to "hybrid" (terminology consideration);
- The choice of weight metrics, it should be decided by the gradients of two part. Do some numerical analysis;
- Plot the gradients changing with respect to weight parameters;
- Plot the whole framework picture of the system;
- Think about the title of two parts what xxx framework and what xxx training algorithm;
- Consider the legends of figures;
- Exist some nadir point for RMSE and MAPE;
- Identify the gap and fill the gap in the organizing;
- Point out the limits of systems, and then turn into introducing the main methodology of this paper;
- In each section, we should introduce the overall parts of this work, section start introducing and end introducing.

Introduction schemes:

1) Background;
2) Electricity prices prediction and battery storage;
3) End to end stochastic optimization;
4) Open source coding;

Smart grid has witnessed the burgeoning of prediction methods and techniques from machine learning domain for renewables and electricity prediction. With the transition to low carbon energy society, the renewables is replacing conventional fossil energies to taking up a larger proportion in generating side. Due to the uncertainty and fluctuation of renewables, a large amount of storage are introduced to the systems.

Renewables are located in various places and feature prominent fluctuation and generation cost, which leads to the fluctuation of electricity prices in retail markets. To minimize the operation cost, accurate electricity price prediction is becoming important to making scheduling for economic dispatching and unit commitment in the system.

REFERENCES

[1] H. Chitsaz, P. Zamani-Dehkordi, H. Zareipour, and P. P. Parikh, "Electricity Price Forecasting for Operational Scheduling of Behind-the-Meter Storage Systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6612–6622, 2018.

[2] L. Peng, S. Liu, R. Liu, and L. Wang,"Effective long short-term memory with differential evolution algorithm for electricity price prediction," *Energy*, vol. 162, pp. 1301–1314, Nov. 2018.

[3] J. Arteaga and H. Zareipour, "A Price-Maker/Price-Taker Model for the Operation of Battery Storage Systems in Electricity Markets," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6912–6920, 2019.

[4] S. Comello and S. Reichelstein, "The emergence of cost effective battery storage," *Nature Communications*, vol. 10, no. 1, p. 2038, May 2019.

[5] P. Donti, B. Amos, and J. Z. Kolter, "Task-based end-to-end model learning in stochastic optimization," *Advances in Neural Information Processing Systems*, 2017, pp. 5484–5494.

[6] B. Wilder, B. Dilkina, and M. Tambe, "Melding the Data-Decisions Pipeline: Decision-Focused Learning for Combinatorial Optimization," *AAAI*, vol. 33, no. 01, pp. 1658–1665, Jul. 2019.

[7] A. N. Elmachtoub and P. Grigas, "Smart 'Predict, then Optimize,' " *Management Science* [online], doi: 10.1287/mnsc.2020.3922.

[8] Z. Chen, L. Wu and Y. Fu, "Real-Time Price-Based Demand Response Management for Residential Appliances via Stochastic Optimization and Robust Optimization," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1822-1831, Dec. 2012.

[9] X. Chen, E. Dall'Anese, C. Zhao and N. Li, "Aggregate Power Flexibility in Unbalanced Distribution Systems," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 258-269, Jan. 2020.

[10] J. Mandi, E. Demirovic, P. J. Stuckey, and T. Guns, "Smart Predict-and-Optimize for Hard Combinatorial Optimization Problems", *AAAI*, vol. 34, no. 02, pp. 1603-1610, Apr. 2020

[11] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems, 2019, pp. 8024–8035. [Online].

[12] PJM., https://dataminer2.pjm.com/list [Online].

[13] Rencher, Alvin C.; Christensen, William F. "Methods of Multivariate Analysis". John Wiley & Sons., 2012 p. 155.

[14] F. Pedregosa et al.,"Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.