Data-driven Locational Marginal Price Prediction From Market Participants' Perspective

by

Zhongxia Zhang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2023 by the
Graduate Supervisory Committee:

Meng Wu, Chair
Vijay Vittal
Lalitha Sankar
Yang Weng

ARIZONA STATE UNIVERSITY

August 2023

ABSTRACT

Energy market participants' optimal bidding strategies and profit maximization is built upon accurate locational marginal price (LMP) predictions. In wholesale electricity markets, LMPs are strongly spatio-temporal correlated. Without access to confidential information on system topology, model parameters, or market operation conditions, market participants can only accept data-driven methods to utilize publicly available market data to predict LMPs. Most previous data-driven studies on LMP forecasting only leveraged temporal correlations among historical LMPs, and very few of them learned the spatial correlations to improve forecasting accuracy. In this dissertation, unsupervised data-driven approaches are proposed to predict LMPs in real-world energy markets from market participants' perspective. To take advantage of the spatio-temporal correlations, a general data structure is introduced to organize system-wide heterogeneous market data streams into the format of market data 2-dimensional (2D) arrays and 3-dimensional (3D) tensors. The system-wide LMP prediction problem is formulated as a sequence prediction problem. A generative adversarial network (GAN) based prediction model is adopted to learn the spatio-temporal correlations among historical LMPs preserved in the market data 3D tensors, then predict future system-wide LMPs. Multi-loss functions are introduced to assist the adversarial training procedure. A convolutional long-short-term memory (CLSTM)-based GAN is developed to improve forecasting accuracy.

All LMP price components are jointly determined by the interactions between the market clearing process and the generator bidding process. The market participants' LMP forecasting problem can be formulated as a sequential decision-making model considering the interactive market clearing and generation bidding decision-making processes. The spatio-temporal decision transformer is proposed to learn the underlying sequential decision-making model from historical spatio-temporal market data and forecast LMPs as the future actions of these interactive decision-making processes. A two-stage approach is proposed to incorporate historical generation bids into energy price prediction from market participants' perspective. Historical generation bids are taken as the first stage's output and the second stage's input in the training process. The implicit correlation among locational bids, demands, and energy prices is learned to improve price forecasting accuracy. The proposed approaches are verified through case studies using both real-world and simulated data.

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my advisor, Dr. Meng Wu, for providing exceptional support and guidance during my research journey. Her invaluable expertise, patience, and encouragement have played a pivotal role in shaping the success of my doctoral work.

Furthermore, I would also like to extend my thanks to my esteemed committee members, Dr. Vijay Vittal, Dr. Lalitha Sankar, and Dr. Yang Weng, for their insightful feedback and valuable contributions to my research. Their expertise and guidance have been instrumental in enhancing the quality and rigor of my work.

Lastly, I am deeply grateful to my parents and all my family members for their unwavering love, support, and understanding. Their constant encouragement and belief in me have been a source of motivation and inspiration throughout my doctoral journey.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

Table Page

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

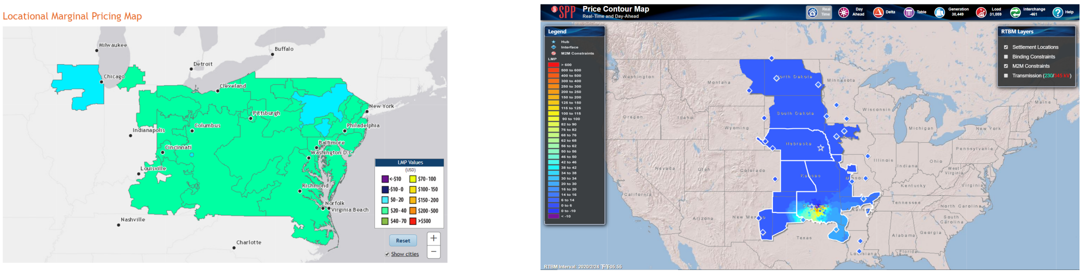
INTRODUCTION

1.1 Background

Driven by the increasing fuel price and rapid growth of renewable energy generation, electricity price is becoming more difficult to be forecasted accurately by market participants. The volatility of electricity prices comes from the uncertainties of demands and the strategic bidding behavior of generators. After the deregulation of the electricity market and the introduction of competitive markets, the traditionally monopolistic and government-controlled electricity markets have been reshaped. In the United States, most electricity is traded in electricity markets run by regional transmission organizations (RTO) or independent system operators (ISO) under competitive market mechanisms using spot and forward contracts [7; 8]. There are seven RTOs and ISOs operating in the United States, serving two-thirds of the population: PJM Interconnection (PJM), Midcontinent ISO (MISO), Electric Reliability Council of Texas (ERCOT), California ISO (CAISO), Southwest Power Pool (SPP), New York ISO (NYISO), and New England ISO (ISO-NE).

All RTO and ISO electricity markets are designed as a multi-settlement system with day-ahead (DA) markets and real-time (RT) markets. The DA markets are cleared before the operating day as a forward market to produce stable prices, whereas the RT markets (also called the balancing market) are cleared on the operating day as a spot market to reconcile any differences between the schedule in the DA market and the RT load. The DA market is responsible for providing generators and load-serving entities a means for scheduling their activities sufficiently prior to their operations, based on a forecast of their needs and consistent with their business strategies. Generation and demand bids scheduled by the DA market are settled by the DA market prices. Because of weather and demand uncertainties, there are always deviations between the scheduled amount of electricity based on the DA forecast and the actual RT load. Therefore, the RT market is cleared to serve the difference between scheduled and actual RT demand and keep the system's reliability. Additional energy offers provided by generators and megawatts over or under-produced relative to the DA commitments are settled at RT prices.

Due to demand uncertainty, transmission congestion, generator forced outages, and other unforeseen events, electricity prices in RT markets are more volatile than the electricity prices in DA markets. Without considering these events, the DA market is usually settled at stable prices with fewer fluctuations and spikes. However, renewable penetration and energy storage encourage energy market participants to schedule and manipulate their generation and demand strategically, which increases the uncertainties and volatility for both DA market prices and RT market prices. Price prediction is critical for energy market participants to develop optimal bidding strategies and maximize their profits. The increasing price volatility caused by the increasing integration of renewable resources makes price prediction more difficult for market participants. Unlike system operators, market participants have no access to confidential market information, such as power grid parameters, topology, and operating conditions, which are essential for precise price prediction. The lack of such confidential market information leads to more challenges for market participants to predict the highly volatile electricity prices. The updated publicly available market data provided by RTOs and ISOs include historical DA prices, RT prices, demands, and partial generation information, such as generation mix data. These data are usually visualized by the colored maps as shown in Fig. 1.1 for each operating time interval. Fig. 1.1a shows a zonal price map from PJM interconnection, where the color in each zone represents the zonal price for this price zone. Fig. 1.1b shows a nodal price map from Southwest Power Pool (SPP), where the color in each node represents the nodal price for this price node. These price maps are updated for each operating time interval. The spatio-temporal correlations among the public market data are represented by a time series of visualized price maps, with can be utilized to predict future prices from market participants' perspective.



(a) Zonal locational marginal price map [2]. (b) Nodal locational marginal price map [4].

Figure 1.1: Colored Price Data Map.

1.2 Locational Marginal Price

In the United States, most electricity markets adopt the concept of locational marginal price (LMP) for both the DA market and the RT market. LMPs for energy cleared in the DA market are calculated on a day-ahead basis for each hour as day-ahead locational marginal prices (DALMP); LMP for the energy cleared in the RT market is calculated every five minutes or fifteen minutes during the operating day as real-time locational marginal prices (RTLMP). The hourly RTLMP is the weighted average of RTLMPs calculated within one hour. Both RTLMP and DALMP are calculated at nodes, load zones, and hubs; LMPs of load zones and hubs are calculated based on a weighted average of nodal LMPs in their area.

DALMPs for all nodes (locations) in the market are determined by the system operator based on the unit commitment and economic dispatch and the prices of energy offers and bids. DALMPs are calculated by applying an optimization to minimize energy cost, given scheduled system conditions, scheduled transmission outages, and any transmission limitations that may exist. To implement this optimization, the system operator estimates the system conditions based on the expected transmission system configuration and the set of offers and bids submitted by market participants. After solving the optimization problem, the cost of serving an increment of load at each node is calculated, including three components: (1) the energy price at which the market participant has offered to supply or consume an additional increment of energy from the resource; (2) the congestion price associated with increasing the output of the resource or reducing consumption of the resource when there is not enough transmission capacity for all of the least-cost generators to be selected; (3) the losses price associated with transmission losses caused by the increment of load and supply. The energy prices for all nodes in the same market are identical, and the losses prices wouldn't vary too much across the market footprint. When there are transmission constraints or congestion, LMPs may vary significantly across the market footprint, which causes the spatial difference of LMPs among the price nodes in the market.

RTLMPs of electricity for all price nodes (locations) in the market are determined by the system operator based on the system conditions described by the power flow solution produced by the state estimator for the pricing interval. RTLMPs are calculated by applying an optimization method to minimize energy cost, given actual system conditions, a set of energy offers and bids, and any binding transmission constraints that may exist. Similar to DALMPs, RTLMPs consist of three components: (1) the energy price; (2) the congestion price; (3) the losses price. During the operating day, the RTLMP calculation is performed every five-minute or fifteen-minute interval, producing a set of nodal RTLMPs based on system conditions during the pricing interval. Then, the RTLMPs produced at pricing intervals during an hour are integrated to determine the nodal RTLMPs for that hour. The RT market will clear the difference between DA commitments and the actual RT demand based on separate, second RTLMPs, which are either paid or charged to participants in the DA market for demand or generation that deviates from the DA commitments [9; 10]. Compared to the DA market, the RT market has a smaller market volume. There is an increased likelihood of supply and demand imbalances in the RT market, which lead to both positive and negative price movements. Therefore, RTLMPs have more volatility and more spikes and are more difficult to predict.

For a fixed RT electricity market, the RTLMP uncertainties are caused both by demand uncertainties and supply uncertainties. In the following subsections, we demonstrate these uncertainties by outlining the basic concepts of LMPs with the DC optimal power flow (DCOPF) formulation.

1.3 Spatio-Temporal Correlations and Uncertainties of LMPs

LMPs are dual variables derived from solving the optimal power flow problem. In general, a DCOPF [11; 12] is widely used by system operators to calculate LMPs, which can be formulated as the following optimization problem:

$$\underset{\mathbf{P}}{\text{minimize}} \quad \sum_{i=1}^{N} a_i P_i^2 + b_i P_i + c_i \tag{1.1a}$$

$$\text{subject to} \quad \sum_{i=1}^{N} P_i - \sum_{i=1}^{N} L_i = 0 : \lambda, \tag{1.1b}$$

$$\mathbf{P}^- \leq \mathbf{P} \leq \mathbf{P}^+ : \sigma^-, \sigma^+, \tag{1.1c}$$

$$\mathbf{F}^- \leq \mathbf{T}(\mathbf{P} - \mathbf{L}) \leq \mathbf{F}^+ : \mu^-, \mu^+. \tag{1.1d}$$

where $\mathbf{P}$, $\mathbf{L}$ are the vectors of active power generations and loads at N buses, respectively. P_i, L_i are i^{th} elements in $\mathbf{P}$ and $\mathbf{L}$, respectively. $\mathbf{P}^-$, $\mathbf{P}^+$, $\mathbf{F}^-$ and $\mathbf{F}^+$ are the vectors of upper and lower limits for power generations $\mathbf{P}$ and transmission line flows $\mathbf{F} = \mathbf{T}(\mathbf{P} - \mathbf{L})$, respectively. $\mathbf{T}$ is the power transfer distribution factor (PTDF) matrix. λ, σ^-, σ^+, μ^- and μ^+ are the Lagrange multipliers of the corresponding constraints. For each generation cost function $C_i = a_i P_i^2 + b_i P_i + c_i$, a_i, b_i and $c_i \in \mathbb{R}$ are the generation bidding variables submitted by the generator owner at bus i. These generation bidding variables may vary frequently instead of being constant.

In (1.1d), with the PTDF matrix $\mathbf{T}$ commonly used in commercial DCOPF software, the active power flows over transmission lines are represented as a linear combination of nodal power injections. The PTDF matrix $\mathbf{T}$ describes this linear mapping, and can be written as [11]:

$$\mathbf{T} = [0\mathbf{DAB}^{-1}] \tag{1.2}$$

where matrices $\mathbf{D}$, $\mathbf{A}$, $\mathbf{B}$ describe power grid topological and physical properties [11]. Given this DCOPF formulation, the vector of nodal LMPs, $\mathbf{LMP}$, can be represented as [11]:

$$\mathbf{LMP} = \frac{\partial \mathcal{L}}{\partial \mathbf{L}} = \lambda 1 + \mathbf{T}^T(\mu^- - \mu^+) \tag{1.3}$$

where $\frac{\partial \mathcal{L}}{\partial \mathbf{L}}$ is the partial derivative of the DCOPF's Lagrangian function $\mathcal{L}$. Assuming constant grid topology and physical properties in $\mathbf{T}$, LMPs are the linear combination of Lagrange multipliers λ, μ^- and μ^+. λ is associated with the supply-demand equality constraint in (1.1b). μ^+ and μ^- are associated with line flow limit constraints in (1.1d). It is clear that λ is determined by the total load $\sum_{i=1}^{n} L$ and the bidding variables submitted by generators spatially dispersed across the grid. Given total load $\sum_{i=1}^{n} L$, μ^+ and μ^- are determined by geographical distributions of loads across the grid. Therefore, LMPs across the grid are spatially correlated through the PTDF matrix and temporally correlated through hourly load/generation variations. Consequently, the spatio-temporal uncertainties in both demands and supplies will lead to LMP uncertainties.

1.4 Report Organization

The rest of the report is organized as follows: Chapter 2 provides a literature review of the existing works on electricity price prediction, including model-based approaches and model-free approaches.

Chapter 3 describes a general data structure, including daily and hourly structures, for market data 2D matrices (images) and market data 3D tensors (videos) and a formulation of the price prediction problem. The system-wide spatio-temporal correlated market data is organized into a 3D tensor. The historical market data tensor stores spatio-temporal correlations among system-wide LMPs. The next time step LMP prediction problem is formulated as a problem of generating a future 2D array, given a historical 3D tensor.

Chapter 4 introduces a convolutional generative adversarial network (GAN)-based prediction model for both day-ahead and hour-ahead forecasts of system-wide LMPs. The proposed GAN-based model is trained using multiple loss functions during the adversarial training procedure. The feature selection and calibration models are implemented to improve price prediction accuracy.

Chapter 5 introduces a convolutional long short-term memory (CLSTM)- based GAN model. The LMP prediction problem is formulated as a spatio-temporal sequence-to-sequence forecasting problem. The previous GAN model in Chapter 4 is updated by replacing the convolutional networks with CLSTM networks.

Chapter 6 introduces a stacked decision transformer-based approach for the spatio-temporal forecasting of RTLMPs. The RTLMP forecasting problem is formulated as learning the implicit sequential decision making process of generator bidding and RT market clearing without any knowledge of generators' bidding strategies or system operators' OPF model. The sequential decision making process is learned using limited public market data.

Chapter 7 introduces a two-stage approach to incorporate historical generation bids into energy price prediction from market participants' perspective. The implicit correlation of locational bids, demands, and energy prices is learned to improve price forecasting accuracy.

Chapter 8 concludes this report and provides the potential for the future work of this study.

Chapter 2

LITERATURE SURVEY

Existing LMP prediction methods include two groups: (1) forecasting from system operators' perspective; (2) forecasting from market participants' perspective. All these LMP forecasting methods can be classified into five main categories: simulation models, gaming theory models, statistical models, machine learning models, and hybrid models.

2.1 Simulation Models

Simulation methods predict energy prices by simulating the operation of power systems based on the generation models and transmission models [13; 14]. The actual generation dispatch with system topology, network parameters, and operating conditions are taken into consideration by simulation methods from the system operators' perspective. With full knowledge of detailed system operation data and power network models, the mathematical model is established to solve the security-constrained optimal power flow (SCOPF) in order to generate LMP forecasts for every price node across the entire system.

Market assessment and portfolio strategies (MAPS) algorithm developed by GE Power Systems Energy Consulting [14] is used to predict the hour-by-hour market dynamics while simulating the transmission constraints in the power system. MAPS takes the detailed load, transmission, and generation units' data as input to generate complete unit dispatch, LMPs at all generation and load buses, and power flow in each transmission line as outputs. UPLAN software [15], developed by LCG consulting, solves multi-commodity, multi-area optimal power flow (MMOPF) and performs

Monte Carlo simulation to forecast electricity prices. It simulates the participants' bidding strategy in different markets, such as the energy market, ancillary service market, and emission allowance market. Compared to the MAPS model, UPLAN takes additional information of price drivers as input, such as hydrological conditions, fuel prices, and competitive bidding behaviors, to forecast the probabilistic price distribution across different energy markets. Both these two methods are complicated to implement, and the computational cost is very high.

To reduce the computational cost, multiparametric programming-based simulation method [16] and online simulation method [17] are applied to forecasting LMP from the operators' perspective. The multiparametric programming [16] is formulated based on the conditional probability mass function of the RTLMPs estimated by Monte Carlo simulation. This methodology incorporates uncertainty models such as load and stochastic generation forecast and system contingency models. The online simulation method in [17] solves multiparametric programming with the inputs of power flows, dispatch levels, and system states at the time of forecasting based on online dictionary learning.

All these simulation-based methods require perfect knowledge of system models, such as system topology and physical properties described in PTDF matrix $\mathbf{T}$, which is not shared with market participants. Market participants cannot take advantage of the above simulation approaches without confidential system model information and market data.

2.2 Game Theory Models

The simulation models mentioned above usually do not consider market participants' bidding variants. In reality, market participants in the energy market intend to adjust their bidding curves from the actual marginal costs of their generation to maximize their revenue. The second group of LMP prediction models is game theory models, which are utilized to model the strategic bidding of market participants and find the solution for the game [18]. This group of models does not require any historical market data. The strategic market equilibrium analysis is the key point for gaming theory models.

In [19; 20], Nash-Cournot models of imperfect competition among electricity producers are formulated as mixed linear complementarity problems (LCP) with and without arbitrage on a linearized DC network with affine price functions. The proposed models calculate a market equilibrium (Nash-Cournot equilibrium) for generation and transmission. A market equilibrium is defined as a set of prices, generator outputs, transmission flows, and consumption that satisfy each market participant's first-order conditions for maximization of its profit while clearing the market. Two types of arbitrage are considered in [20]. In the first type of arbitrage, the market participants owning the generator anticipate the effect of arbitrage upon prices at different locations (Stackelberg assumption). In the second type of arbitrage, the market participants take the arbitrage quantities as inputs in their problems (Cournot assumption), and the arbitrager solves a separate profit maximization problem that takes the electricity prices and the transmission costs as inputs.

Considering strategic bidding and load elasticity, [21] presents a fundamental bid-based stochastic model forecasting hourly prices and average prices over a specified time interval. Taking the uncertainty of the unit availability and demand into consideration, this model captures both the economic and physical aspects of the pricing process based on Bertrand, Cournot and supply function equilibrium (SFE). In [21], the model can capture the stochastic dependence of the price on costs, load, reliability, and bidding strategies. However, the factors related to transmission congestion, transmission outages, and unit commitment are ignored by this model, which makes it focus more on qualitative rather than quantitative results. [21] concludes that there will remain no alternative other than using Monte Carlo methods if unit commitment and transmission constraints are included in the system model.

2.3 Statistical Models

Statistical models employ a mathematical combination to forecast the prices using previous prices and/or previous or current values of exogenous factors, especially demand, generation, and weather variables. Statistical models are widely accepted because it is easier for engineers and system operators to understand their behavior by attaching some physical interpretation to their components [7].

2.3.1 Similar-Day Methods

Similar-day method (Naive method) is a very popular benchmark in the price forecast area because of the simple implementation and good performance in industries. To forecast the current day's prices, it searches historical data for days with characteristics similar to the target day and considers those historical values or the average value of prices in previous similar-days as forecasts of future prices [22; 23; 24]. The similarity between the target forecast day and previous days is evaluated by Euclidean norm with weighted factors. The drawback of similar-day methods is that forecasting procedures without careful calibration fail to pass the 'Naive test' surprisingly often, leading to low accuracy when large variations exist in prices.

2.3.2 Exponential Smoothing Methods (ESM)

ESM is another simple benchmark in the price forecasting area. The forecast is constructed based on an exponentially weighted average of past observations [25; 26;

27]:

$$s_t = \alpha x_t + (1-\alpha)s_{t-1} = \alpha \sum_{j=0}^{t} (1-\alpha)^j x_{t-j}, t>0 \tag{2.1}$$

where $\alpha (0<\alpha<1)$ is the smoothing factor, x_t is the observation at time t, s_t is the smoothed forecast calculated by the weighted average of the current observation x_t, and the previous smoothed statistic s_{t-1}. As shown in equation 2.1, the weights decrease exponentially depending on the value of smoothing factor α. A further complex model has been developed with the independent trend and seasonal component added. The drawback of the ESM method is that the smoothing factors, such as α, are usually difficult to determine or estimate.

2.3.3 Regression Models

Regression is very popular and well-developed in statistical analysis. Multiple regression focuses on learning the relationship between several independent or predictor variables and a dependent or criterion variable. The regression model is fitted by minimizing the sum-of-squares of the differences between observed and predicted values. The basic regression formulation is shown:

$$P_t = \mathbf{B}\mathbf{X}_t + \varepsilon_t = b_1 X_t^{(1)} + \cdots + b_k X_t^{(k)} + \varepsilon_t \tag{2.2}$$

where $\mathbf{B}$ is a $1 \times k$ vector of constant coefficients, $\mathbf{X}_t$ is the $1 \times k$ vector of regressors and ε_t is an error term. This standard formulation is estimated by performing maximum likelihood methods to learn the correlations between price P_t and regressors. Several types of data can be taken as regressors for multiple regression methods, such as prices, available nuclear capacity, temperatures, and rainfall in. In the time-varying regression (TVR) model, the constant coefficients vector $\mathbf{B}$ is replaced by $1 \times k$ time-varying coefficients vector $\mathbf{B}_t$ as shown in equation 2.3:

$$P_t = \mathbf{B}_t\mathbf{X}_t + \varepsilon_t = b_{1,t} X_t^{(1)} + \cdots + b_{k,t} X_t^{(k)} + \varepsilon_t \tag{2.3}$$

Parameters in the TVR model are estimated by performing state space methods, and the Kalman filter [28].

2.3.4 Time Series Models

Time series statistical models, including auto-regression (AR), moving average (MA), autoregression moving average (ARMA) [29], autoregressive integrated moving average (ARIMA) models [30], and generalized autoregressive conditional heteroskedastic (GARCH) [7; 31; 32; 33; 34; 35; 36], are widely accepted in price forecast of the electricity market.

AR model calculates the forecasted price as the sum of the stochastic error and the linear combination of the historical prices. MA model calculates the forecasted price as the sum of the average value and the moving average value of the white noise in the price series. ARMA model is a combination of AR and MA, which assumes that the time series to learn is weakly stationary. ARMA(p,q) model calculates forecasted price as a linear combination of its p past values(AR part) and q previous values of the noise (MA part):

$$\phi(B)X_t = \theta(B)\varepsilon_t \tag{2.4}$$

where B is the backward shift operator, ε_t is i.i.d. noise with zero mean and finite variance. $\phi(B)$ and $\theta(B)$ are defined as follow:

$$\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p \tag{2.5}$$

where $\phi_1, \cdots, \phi_p$ are the coefficients of AR polynomials.

$$\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q \tag{2.6}$$

where $\theta_1, \cdots, \theta_q$ are the coefficients of MA polynomials.

The ARIMA model is a generalization of an ARMA model. ARIMA(p,d,q) consists of three types of parameters:

$$\phi(B)\nabla^d X_t = \theta(B)\varepsilon_t \tag{2.7}$$

where $\nabla^d x_t \equiv (1 - B^d)x_t \equiv x_t - x_{t-h}$ is the lag-d differential operator. In [37], the standard seasonal ARMAX time-series model is generalized to the L^2 Hilbert space to model complex time dependencies of a time series of electricity prices. The proposed model also allows the inclusion of explanatory variables, such as weather data, which are important drivers of electricity price time series.

By adding the moving average of past conditional variances to an autoregressive process, the GARCH is obtained:

$$X_t = \varepsilon_t \sigma_t \tag{2.8}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i h_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{2.9}$$

where σ_t^2 is the conditional heteroscedasticity, α_i and β_i are the coefficients $(\alpha_i, \beta_i \geq 0, \alpha_0 > 0)$. The GARCH model can be combined with other time series models, such as ARIMA-GARCH [38] and ARMAX-GARCH [39].

In [40], a multivariate LASSO statistical model is developed to forecast the electricity prices of eleven European datasets. This method only accounts for the statistical relationship among prices without considering the actual spatio-temporal uncertainties of demands and supply bids.

All the traditional time series models mentioned above can only capture the linear pattern of time series data because they only forecast the price changes from the sequence itself without taking the impact of the market and other environmental factors into consideration. In the real-world electricity markets, especially in RT markets, there are a large number of rapid variations and high-frequency changes, which cannot be successfully captured by this type of model. Therefore, this type of model is more suitable for DA market forecasts with strong low-frequency patterns. Poor performance of spikes prediction is a common problem faced by most statistical methods, especially for price-only models. A lot of spike detection methods, such as recursive filters [41; 42], variable price thresholds [43], fixed price change thresholds [44], regime-switching classification [45], and wavelet filtering [46], have been proposed.

2.4 Machine Learning Models

Data-driven machine learning approaches, such as nearest neighbor models, support vector machines (SVM), and neural networks (NN), have been developed to learn the nonlinear relationship between LMPs and other market factors.

2.4.1 Support Vector Machine

SVM is employed to determine the decision boundaries to classify observed data points into different groups [47]. Once the state space is partitioned by these decision boundaries, the future data points can be classified and predicted. Usually, the observed data points in the original state space can not be separated by a single linear function. Therefore, SVM either performs a non-linear mapping of the data points in the original space into a high-dimensional space or uses a set of hyperplanes to construct nonlinear boundaries. [48] investigates the LMP-Load coupling in security-constrained economic dispatch (SCED). This unique one-to-one mapping is depicted by the concept of system pattern region (SPR) which can be partitioned by SVM. Without knowledge of confidential system information, such as network topology and operating conditions, SPRs can solely be estimated by SVM from historical LMP and load data. However, this method only predicts future LMP ranges instead of specific LMP values. Moreover, this SPR-based method requires predicted nodal load data as inputs for accurate LMP prediction, which is not always available to the public. A holistic approach [1] is deployed to forecast LMPs from market participants' perspective. The holistic approach in [1] can identify grid operating regimes by learning the relationship between generation mix and zonal demand on one hand and RTLMPs on the other hand. Both the SPR-based approach and the identification of grid operating regimes expect to learn a unique mapping from locational loads to LMPs. However, this unique mapping only holds under the assumption of fixed generation bids and operating conditions. In a real-world system with time-varying operating conditions, generators always change their bids strategically. The unique mapping learned by the SRP-based methods can not be guaranteed. With the assumption of no variation of generation bids, SPR-based forecasting methods cannot capture the temporal correlations among historical LMPs. Indeed, these methods are only tested in simulated markets without any generation bidding variations.

2.4.2 Neural Networks

Different types of artificial neural networks (ANN) have been applied to electricity price forecasts, such as feed-forward neural network [49], recurrent neural network (RNN) [50], fuzzy neural network [51] and cascaded neural network [52; 53]. In machine learning, neural networks are utilized to learn the nonlinear relationship between the input and the output variables. Therefore, the nonlinear relationship between the price to forecast and the historical data can be captured by the neural network. RNN is designed to learn the dependencies among input time series. RNNs read in the data at one time step every time and extract the dependence between the input data and a hidden state from the last time step. A set of 'context units' in RNN are responsible for dependency in the historical time series [54]. The extracted dependency will be passed to the next step of the RNN operation as a time-dependent hidden state, which is the internal representation of past events. There are connections from the hidden (middle) layer to these context units; they have fixed weights and do not have to be updated during training. In the general RNN structure, the outputs of all neurons are connected recurrently to all neurons in the network. However, RNN has difficulty in grasping this dependency over long time periods. This limitation is resolved by the long short-term memory (LSTM) network [55], which is recently utilized to improve electricity price forecasting performance. Based on RNN, LSTM is designed to take advantage of an additional hidden state as a memory to keep long-term dependencies from past inputs. LSTM is widely combined with other models to improve electricity price forecasting accuracy. In [56], a hybrid LSTM-deep NN (DNN) structure is proposed to accurately model complex nonlinear electricity price sequences. In [57], an optimized heterogeneous structure LSTM network is constructed to improve the accuracy and stability of the general LSTM model in electricity price forecasting. Before feeding into the LSTM model, the electricity price sequence is decomposed based on the characteristics of non-linear, non-stationary, and multi-frequency superposition of electricity price. Individual LSTM networks are trained based on the non-linearity of high and low-frequency sequences, respectively. In general, these LSTM models have a better performance in forecasting the nonlinearity of electricity prices. However, they ignore the spatial correlations among historical LMPs and demands, which should also be a major concern of LMP forecasting.

The major advantage of data-driven machine learning approaches is their ability to handle complexity and non-linearity. The machine learning methods are better at modeling these nonlinear features of electricity prices than the statistical techniques discussed above. However, the drawbacks of existing machine learning-based approaches are obvious; the data-driven approaches are difficult to interpret physically as most of them are black-box models.

2.5 Hybrid Models

It is impossible to find a universal single method that performs best for all the circumstances [58]. Recently, hybrid models combining two or more existing forecasting methods become popular. The hybrid models can exhibit individual techniques' advantages and significantly improve prediction accuracy. For example, up to 22% overall improvement of forecasting performance by combining methods from different domains has been found in [59]. Machine learning methods and time series statistical forecasting methods are often combined to form effective methods, such as Fuzzy + ARIMA + ANN [60], ARIMA + SVM [61], singular spectrum analysis (SSA) + modified wavelet neural network (WNN) [62], etc.

In [63], a feature selection technique is proposed to improve the accuracy of short-term price forecasting for neural network-based prediction models. In [64; 65; 66], multi-neural network-based models are implemented in price forecasting. The combinatorial neural network [65] is trained by a stochastic search method, and a sliding window is used to incorporate previously predicted values for a multi-period forecast. References [64; 66] focus on forecasting components of LMPs. The final LMP prediction includes individual predicted energy price, congestion price, and loss price. These methods mainly capture the temporal correlations between LMPs and demands, without considering the spatial correlations among system-wide LMPs.

Chapter 3

DATA DESCRIPTION AND PROBLEM FORMULATION

3.1 Data Description

All RTOs and ISOs in the U.S. publish their market data and operation reports for both the DA market and the RT market, which can be utilized in price prediction. These reports include both nodal market data and zonal market data. Publicly available market data provided by PJM, ISO-New England, MISO, and SPP are included in this chapter.

3.1.1 PJM Market Data

PJM includes 38164 price nodes located in 20 different load zones. For each price node, the zip code is provided by PJM based on geographical data. PJM shows a zonal LMP map updated every five minutes, as shown in Fig. 3.1. In Fig. 3.1, each zone is filled with one color determined by its zonal price at that moment. Besides LMP data, load data and generation-related data on the operating day are also provided. The publicly available data is summarized in Table. 3.1.

3.1.2 ISO-New England Market Data

ISO-NE market includes 1199 price nodes located in 8 different load zones. Similar to PJM, ISO-NE shows a zonal LMP updating every five minutes, as shown in Fig. 3.2. Besides LMP data, load data, dispatch fuel mix data, and temperature data on an operating day are also provided by ISO-NE. The generation bid data received by the ISO-NE from participants in the DA energy market and during the re-offer period for

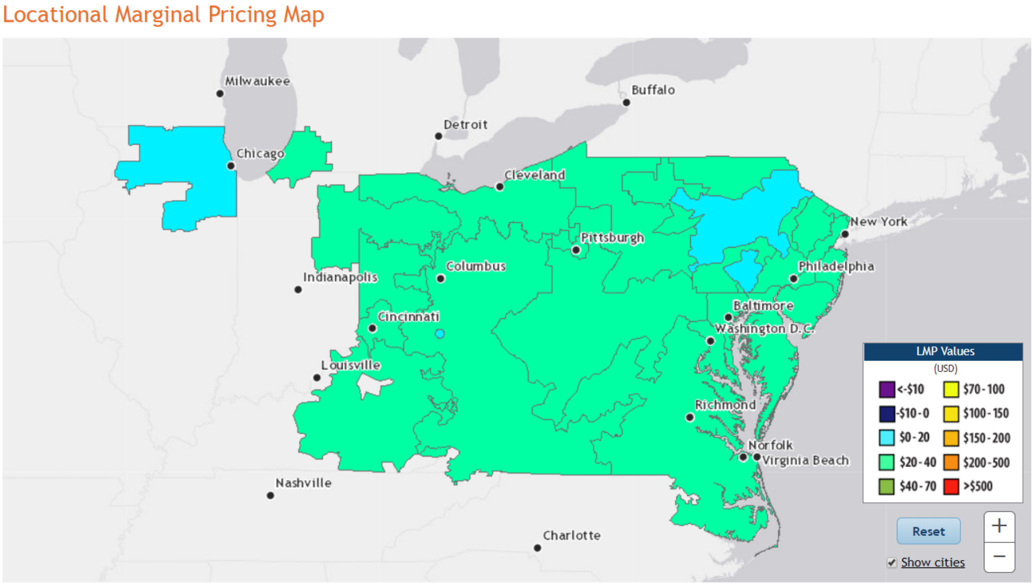


Figure 3.1: PJM LMP Map [2].

consideration in the RT energy market is published monthly on the first day of the fourth month following the operating month, per the FERC-ordered lag period. The publicly available data is summarized in Table. 3.2.

3.1.3 SPP Market Data

SPP market includes 2863 price nodes located in 16 different load zones. Unlike PJM and ISO-NE, SPP shows nodal LMP map updating every five minutes as shown in Fig. 3.3. In Fig. 3.3, the color of every point is determined by the price node's value at that location for that moment.

Besides LMP data, load data and generation mix data in an operating day are also provided by SPP. The publicly available data is summarized in Table. 3.3.

Table 3.1: Public Market Data During an Operating Day in PJM

Data type	**Market**	**Time interval**	**Level**
LMP	RT, DA	five minutes, hourly	zonal, nodal
System energy price	RT, DA	five minutes, hourly	zonal, nodal
Congestion price	RT, DA	five minutes, hourly	zonal, nodal
Marginal Loss Price	RT, DA	five minutes, hourly	zonal, nodal
Load	RT	hourly	zonal
Generation by fuel type	RT	hourly	system
Solar generation	RT	hourly	zonal
Wind generation	RT	hourly	zonal

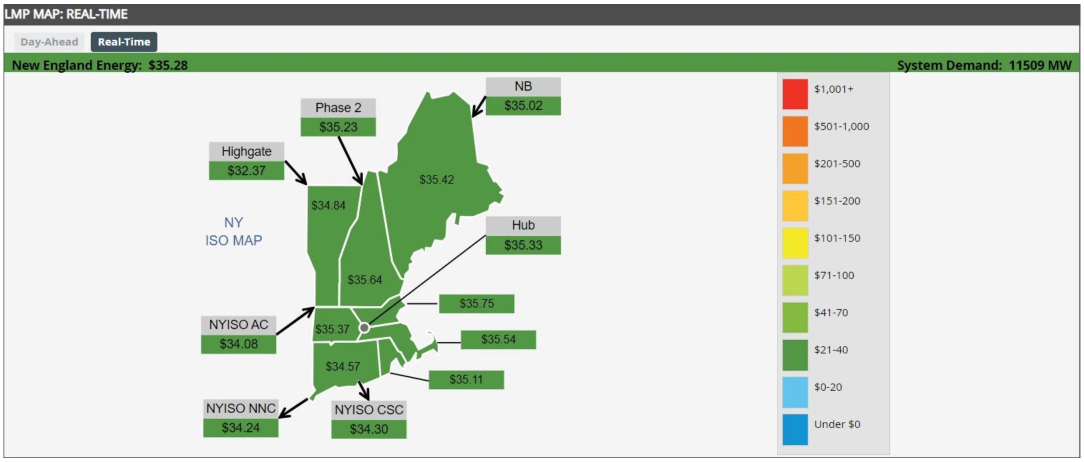


Figure 3.2: ISO-NE LMP Map [3].

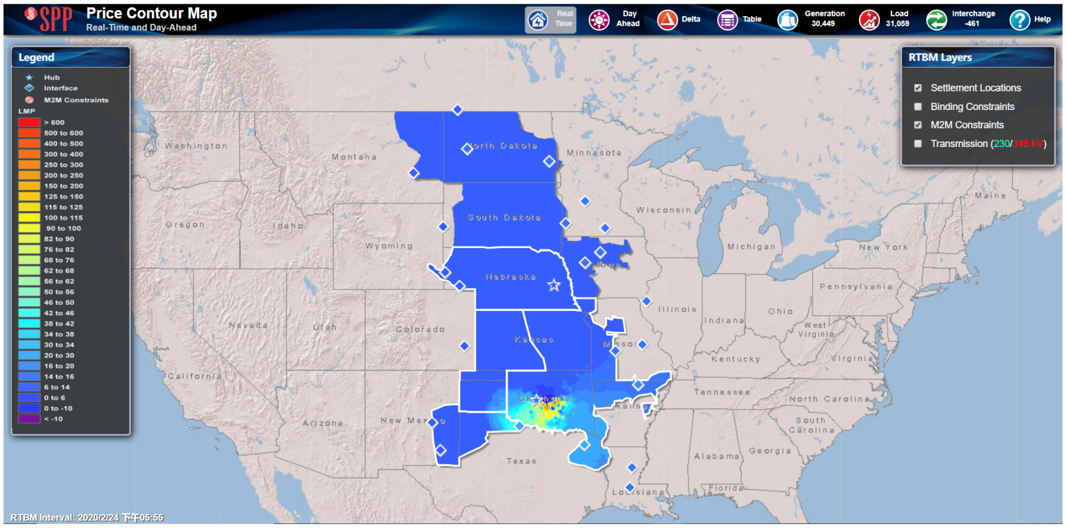
3.1.4 *MISO Market Data*

The MISO market includes 12 cost allocation zones and 10 local resource zones. MISO reports prices for between 1800 and 2000 nodes. Similar to SPP, MISO provides nodal LMP contour map updating every five minutes as shown in Fig. 3.4.

Besides LMP data, load data, generation mix, and generation bid data in an

Table 3.2: Public Market Data During an Operating Day in ISO-NE

Data type	**Market**	**Time interval**	**Level**
LMP	RT, DA	five minutes, hourly	zonal, nodal
System energy price	RT, DA	five minutes, hourly	zonal, nodal
Congestion price	RT, DA	five minutes, hourly	zonal, nodal
Marginal Loss Price	RT, DA	five minutes, hourly	zonal, nodal
Load	RT, DA	hourly	zonal
Dispatch fuel mix	RT	varying length	system
Dry bulb temperature	RT	hourly	zonal
Dew point temperature	RT	hourly	zonal
Generation bid	RT, DA	hourly	nodal



RTBM Layers

☑ Settlement Locations
☐ Binding Constraints
☑ M2M Constraints
☐ Transmission (230/345 kV)

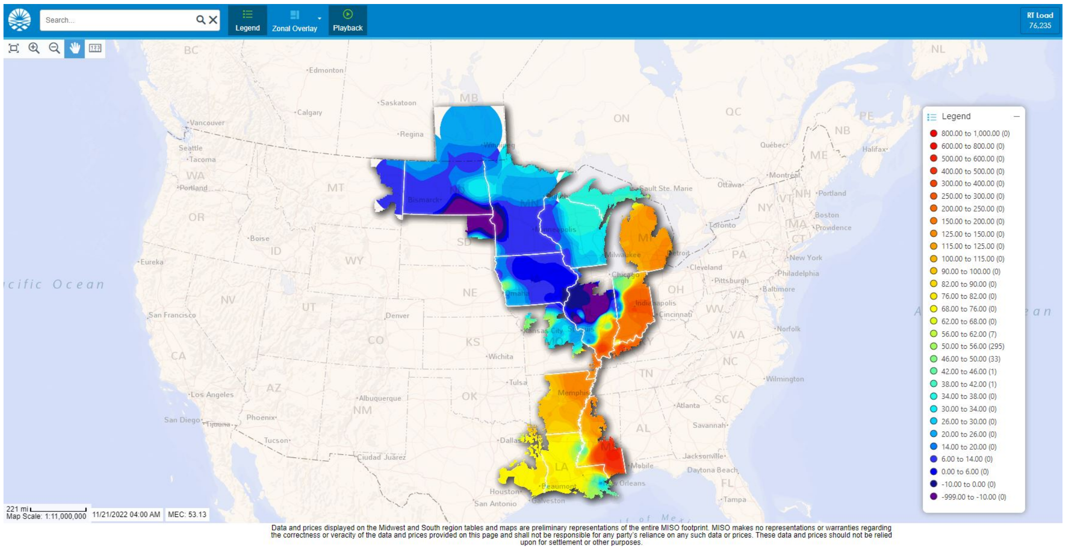
RTBM Interval: 2020/2/24 下午05:55

Figure 3.3: SPP LMP Map [4].

operating day are also provided by SPP. The publicly available data is summarized in Table. 3.4.

Table 3.3: Public Market Data During an Operating Day in SPP

Data type	**Market**	**Time interval**	**Level**
LMP	RT, DA	five minutes, hourly	zonal, nodal
System energy price	RT, DA	five minutes, hourly	zonal, nodal
Congestion price	RT, DA	five minutes, hourly	zonal, nodal
Marginal Loss Price	RT, DA	five minutes, hourly	zonal, nodal
Load	RT, DA	hourly	zonal
Generation mix	RT	hourly	system



Data and prices displayed on the Midwest and South region tables and maps are preliminary representations of the entire MISO footprint. MISO makes no representations or warranties regarding the correctness or veracity of the data and prices provided on this page and shall not be responsible for any party's reliance on any such data or prices. These data and prices should not be relied upon for settlement or other purposes.

Figure 3.4: MISO LMP Map [5]

3.1.5 Data Summary

Almost all markets provide hourly LMP data (system energy price, congestion price, marginal loss price), load data, and generation by fuel type data in an operating day, which can be utilized by a universal prediction approach. Some system operators also publish hourly generation bid data in both the RT market and the DA market,

Table 3.4: Public Market Data During an Operating Day in MISO

Data type	**Market**	**Time interval**	**Level**
LMP	RT, DA	five minutes, hourly	zonal, nodal
System energy price	RT, DA	five minutes, hourly	zonal, nodal
Congestion price	RT, DA	five minutes, hourly	zonal, nodal
Marginal Loss Price	RT, DA	five minutes, hourly	zonal, nodal
Load	RT, DA	hourly	zonal
Generation mix	RT	hourly	system
Generation bid	RT, DA	hourly	nodal

with 3 months delay.

3.2 Data Preprocessing

As shown above, all market operators visualize either zonal or nodal price data using a map for each interval. In the LMP maps, system-wide LMPs are demonstrated using color-coded values according to their geographical locations. Therefore, the LMP maps store spatial correlations among the LMPs at different locations. Considering each LMP map as an LMP image, a series of time-stamped LMP images can be concatenated to form an LMP video. This LMP video stores both spatial and temporal correlations among LMPs in the market. With these representations, the LMP prediction problem can be formulated as a problem of predicting the future frame of the LMP video. Then, the state-of-art video prediction approaches can be applied to predict LMPs by learning the spatio-temporal correlations among historical data.

3.2.1 The Motivating Example: Visualizing LMPs as Images and Videos

Visualization generated using RTLMPs (from 1:00 AM to 11:00 AM on 5/15/2019) obtained from 56 price nodes within the territory of Atlantic Electric Power Company (i.e., the AECO price zone) in PJM Interconnection [2] is shown in Fig. 3.5.

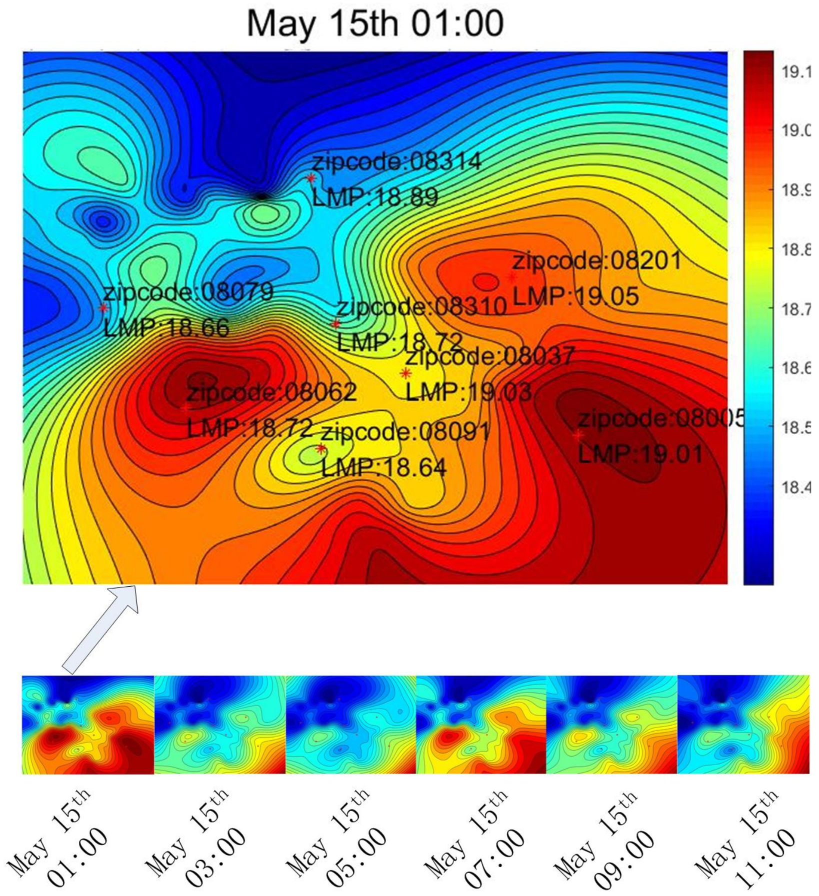


Figure 3.5: RTLMP Rtlmp Visualization of AECO Price Zone in PJM Market from 1:00 AM to 11:00 AM on May 15, 2019.

In Fig. 3.5, eight price nodes are explicitly identified on the image at 1:00 AM with their RTLMP values and corresponding zip codes. The coordinates of the price nodes represent their geographical locations in the system. At the bottom of Fig. 3.5, six images are generated using system-wide RTLMPs at six different hours. The colors on these six images vary both spatially (within the same image) and temporally (between consecutive images), indicating the spatio-temporal variations of RTLMPs. This example clearly shows the spatio-temporal correlations among system-wide RTLMPs can be captured by the spatio-temporal color variations across a series of time-stamped images. This series of time-stamped images then form a video consisting of the above RTLMP visualizations.

The smooth RTLMP visualizations in Fig. 3.5 are generated using biharmonic spline interpolation [67]. It's an intuitive example to show the capability that LMP images and LMP videos can store the spatio-temporal correlations among LMPs. However, in the RTLMP image, the color of 56 points is determined directly by the RTLMP values. The color of the rest area in Fig. 3.5 is determined by the interpolation function which has no physical representation. In Fig. 3.6, two different interpolation techniques (biharmonic spline interpolation for Fig. 3.6(a) and nearest-neighbor interpolation for Fig. 3.6(b)) are applied to an identical dataset (RTLMPs at 56 price nodes at 1:00 AM on 5/15/2019, in AECO price node). The nearest neighbor interpolation results in a less smooth RTLMP visualization with exactly 56 different color zones. Each color zone corresponds to the RTLMP value of a particular price node. These 56 color zones in Fig. 3.6(b) are then re-organized to generate the image in Fig. 3.6(c), with 56 colored squares of the same size. The color of each square is fully determined by the RTLMP value at the corresponding price zone. These colored squares represent pixels in a colored digital image, which are the smallest addressable element in an image. A full comparison between three visualization methods is shown in Fig. 3.7.

Compared with smooth RTLMP visualization in Fig. 3.5, the colored digital images and digital videos consisting of digital images in Fig. 3.6(c) keep the same spatio-temporal correlations among LMPs as while as reduce the computational requirement

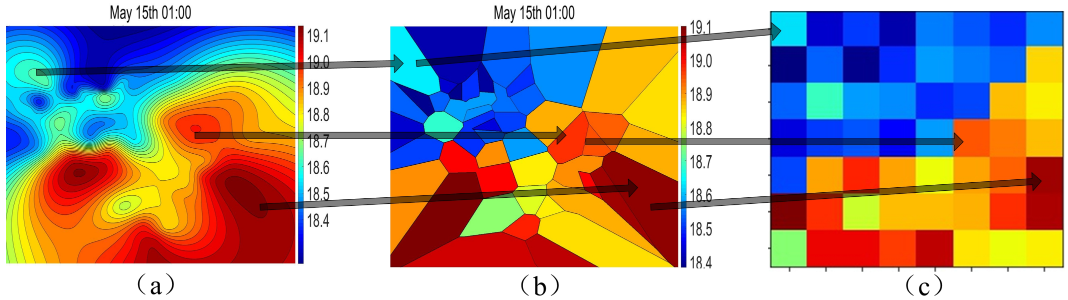


Figure 3.6: Comparison Between RTLMP Visualizations Generated Using Different Interpolation Techniques.

due to their small size. In the following section, we formally define data structures for colored digital pixels (data points), images (2D arrays), and videos (3D tensors).

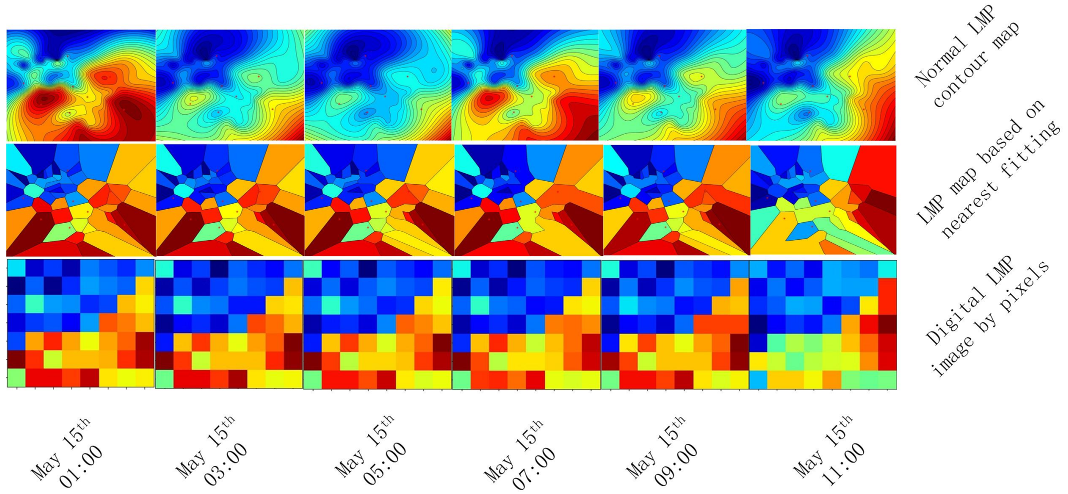


Figure 3.7: Full Comparison Between RTLMP Visualizations Generated Using Different Interpolation Techniques.

3.2.2 Data Structure for Digital Images and Videos

Consider a set of publicly available historical hourly market data collected from $N = h \times w$ different price nodes (locations) for T consecutive hours. There are M types of historical market data for each price location (i.e., the RTLMPs, the DALMPs, the local demand, the distributed energy resources outputs, etc.). All these locational market data can be organized into a 3D tensor as shown in Fig. 3.8.

Definition 1 *Let* $x^t_{i,j}$ *represent a historical market data point collected from* k^{th} *price node at time* t*, where* $k = h \times (i-1) + j$, $k \in [1, N]$*. This market data point includes* M *channels. For example, when* $M = 3$, $x^t_{i,j} = \{x^{t-rtlmp}_{i,j}, x^{t-dalmp}_{i,j}, x^{t-demand}_{i,j}\}$ *with the RTLMP, DALMP, and demand data channels, respectively.*

Definition 2 *At each time* t*,* N *historical market data points collected over the market are arranged into a 2D array* $X^t \in \mathbb{R}^{h \times w}$*. The positions of each market data point in the 2D array are fixed and determined according to their geographical location in the electricity market footprint.*

Definition 3 *Let a 3D tensor* $\mathcal{X} \in \mathbb{R}^{h \times w \times T}$ *represent the set of publicly available historical hourly market data. This tensor* $\mathcal{X}$ *is comprised of* T *2D arrays* X^t*, where* $t \in [1, T]$*.*

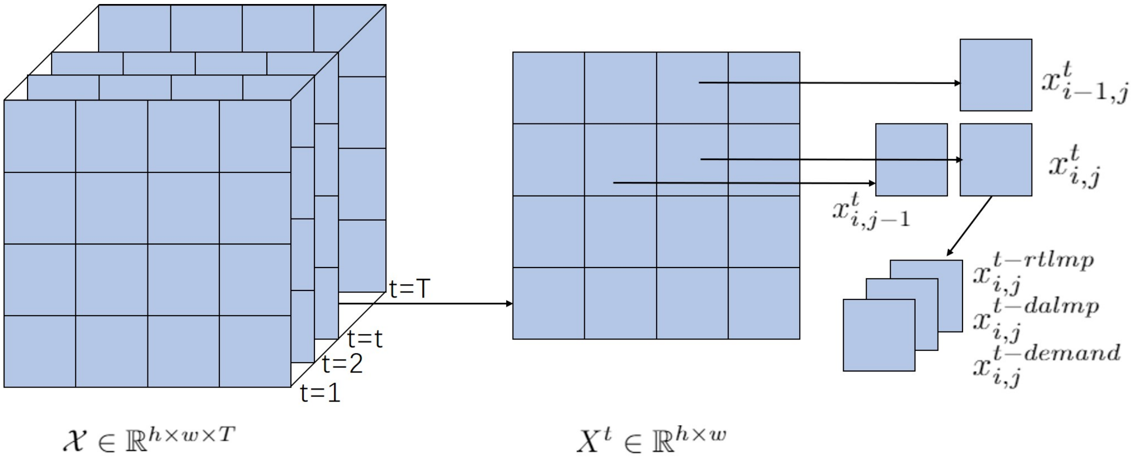


Figure 3.8: Historical Market Data Structure.

As shown from right to left in Fig. 3.8, different types of market data collected from the same location at one timestamp are organized into one historical market data point with different data channels. Then, the historical market data points of all locations at the same time step are organized into one 2D array. Finally, a series of time-stamped 2D arrays are organized into a 3D tensor. The above definitions introduce a general market data structure to conveniently store the spatio-temporal correlations among historical market data in the inter-dependencies along multiple dimensions of the 3D tensor. For example, the correlations among different positions in one 2D array X^t, such as $x^t_{i-1,j}$, $x^t_{i,j}$ and $x^t_{i,j-1}$, represent spatial correlations among historical market data collected from different price nodes (locations) at time t; the correlations among the same positions obtained at different times, such as $x^{t-1}_{i,j}$, $x^t_{i,j}$, $x^{t+1}_{i,j}$, represent temporal correlations among historical market data collected from the same price node (location) at different hours.

In **Definition 2**, each 2D array consists of market data collected at one time (hour). Therefore, the third dimension of the tensor in **Definition 3** represents hours. It can also be modified to represent days. Consider the same set of publicly available historical hourly market data collected from $N = h \times w$ price nodes (locations) for D days. Within each day, we have 24 consecutive 2D arrays (X^1 to X^{24}) as defined in **Definition 2**. They can be reorganized into an enlarged 2D array $X^d_{Day} \in \mathbb{R}^{4h \times 6w}$, where $d \in [1, D]$. In this way, the hourly tensor $\mathcal{X}$ is reshaped to a daily tensor $\mathcal{X}_{Day} \in \mathbb{R}^{4h \times 6w \times D}$, as shown in Fig. 3.9. In this daily tensor, the spatio-temporal correlations among hourly market data within the same day are stored in each day's 2D array X^d_{Day}. The daily market pattern is represented by the correlations among different daily 2D arrays in one daily 3D tensor.

The daily and hourly tensor structures can be adapted to perform day-ahead and hour-ahead predictions of system-wide LMPs, respectively. In the rest of this paper, the hourly tensor structure in **Definition 3** is mainly used to present the prediction approach. In the case studies, both hourly and daily tensor structures are utilized in

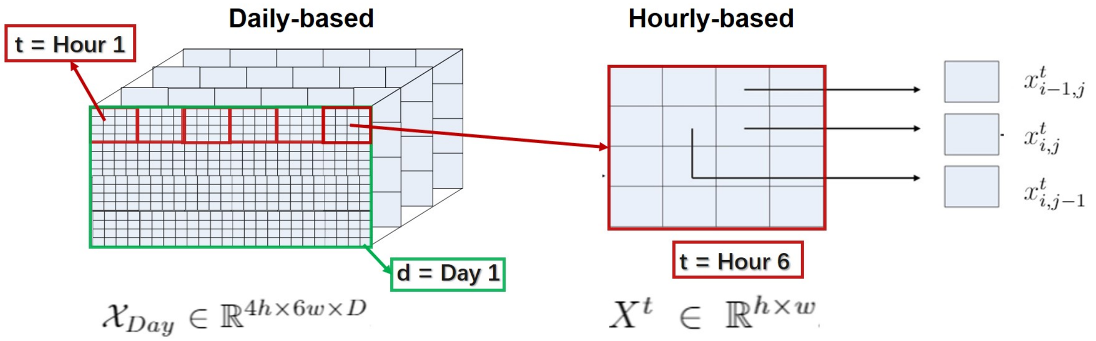


Figure 3.9: The Daily Tensor Structure Reshaped from the Hourly Tensor.

developing the hour-ahead and day-ahead LMP predictors.

3.2.3 Normalization of Historical Market Data

Because each data point in tensor $\mathcal{X}$ contains M channels. The units and scales of these M channels are different. Even for the same channel, the statistics of historical market data may differ year by year. Therefore, all historical data are preprocessed to the range of -1 and 1 by normalization.

After organizing historical locational market data into tensor $\mathcal{X}$, each value $x_{i,j}^{t-ch}$ of the channel ch is normalized as:

$$x_{i,j}^{t-ch^{norm}} = \frac{ln(x_{i,j}^{t-ch^{+}}) - ln(max(\mathcal{X}^{ch^{+}}))/2}{ln(max(\mathcal{X}^{ch^{+}}))/2} \tag{3.1}$$

where

$$x_{i,j}^{t-ch^{+}} = x_{i,j}^{t-ch} - min(\mathcal{X}^{ch}) + 1 \tag{3.2}$$

$$\mathcal{X}^{ch^{+}} = \{X^{1-ch^{+}}, \cdots, X^{t-ch^{+}} \cdots, X^{T-ch^{+}}\} \tag{3.3}$$

where $x_{i,j}^{t-ch^{norm}}$ is the normalized value of $x_{i,j}^{t-ch}$ in channel ch; $x_{i,j}^{t-ch^{+}}$ denotes $(i,j)^{th}$ element of $X^{t-ch^{+}}$; $X^{t-ch^{+}}$ is t^{th} 2D array in tensor $\mathcal{X}^{ch^{+}}$; $\mathcal{X}^{ch}$ is the channel ch of tensor $\mathcal{X}$; $max(\mathcal{X}^{ch^{+}})$ is the largest element in $\mathcal{X}^{ch+}$; $min(\mathcal{X}^{ch})$ is the smallest element of channel ch in tensor $\mathcal{X}$. After normalization, values in the normalized tensor $\mathcal{X}^{norm}$, such as normalized prices and demands, lie within -1 and 1. Equations (3.1)-(3.3) define an one-to-one mapping between $\mathcal{X}$ and $\mathcal{X}^{norm}$.

3.2.4 Market Data Images and Videos

Consider a wholesale electricity market with $m \times n$ price nodes. A set of historical market data of various types (such as RTLMPs, DALMPs, demands, generations, temperatures, etc.) can be collected at each price node. Let $x_{i,j}^{norm}(t) = [v_{i,j}^{norm-1}(t), v_{i,j}^{norm-2}(t), v_{i,j}^{norm-3}(t)]$ be a 1×3 vector containing three types of normalized market data obtained from the $(i,j)^{th}$ price node at time t. Let $X^{norm}(t)$ be a $m \times n$ matrix whose $(i,j)^{th}$ element is $x_{i,j}^{norm}(t)$. According to the data structures defined for images and videos, $X^{norm}(t)$ can be viewed as a colored digital image with a resolution of $m \times n$, and $x_{i,j}^{norm}(t)$ can be viewed as the $(i,j)^{th}$ pixel of this colored digital image. By concatenating a series of such market data images, we obtain a tensor $\mathcal{X}^{norm} = \{X^{norm}(1), \cdots, X^{norm}(t), \cdots, X^{norm}(T)\}$, which can be viewed as a colored digital video containing three different types of historical market data obtained at $m \times n$ different price nodes for time interval $[1, T]$.

Fig. 3.10 shows a market data video consisting of 24 hourly market data images/frames generated using historical data from 56 (7×8) price nodes in the AECO price zone on May 15th, 2019. Each square in Fig. 3.10 represents a pixel whose red, green, and blue color codes take the value of normalized RTLMP, real power demand, and temperature at the corresponding price node, respectively. The color of each pixel is fully determined by the corresponding market data values (after normalization) as shown in Fig. 3.11. It is clear the spatio-temporal variations of the pixel colors represent the spatio-temporal variations of RTLMPs, real power demands, and temperatures during these 24 hours across the AECO price zone. If a learning model is trained to learn the spatio-temporal color variations in the historical market data video, this model could then be applied for system-wide LMP predictions.

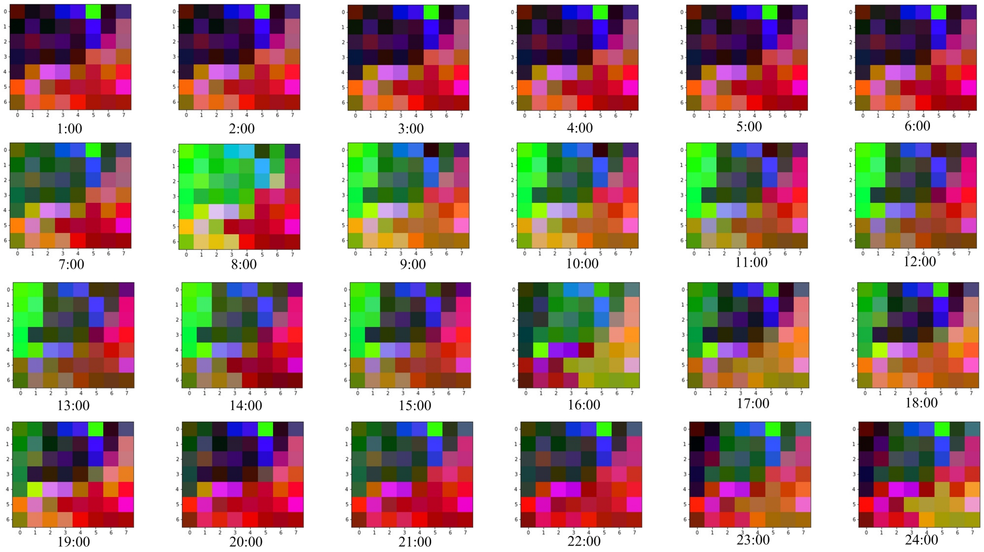


Figure 3.10: Colored Digital Market Data Video of AECO Price Zone in PJM Market.

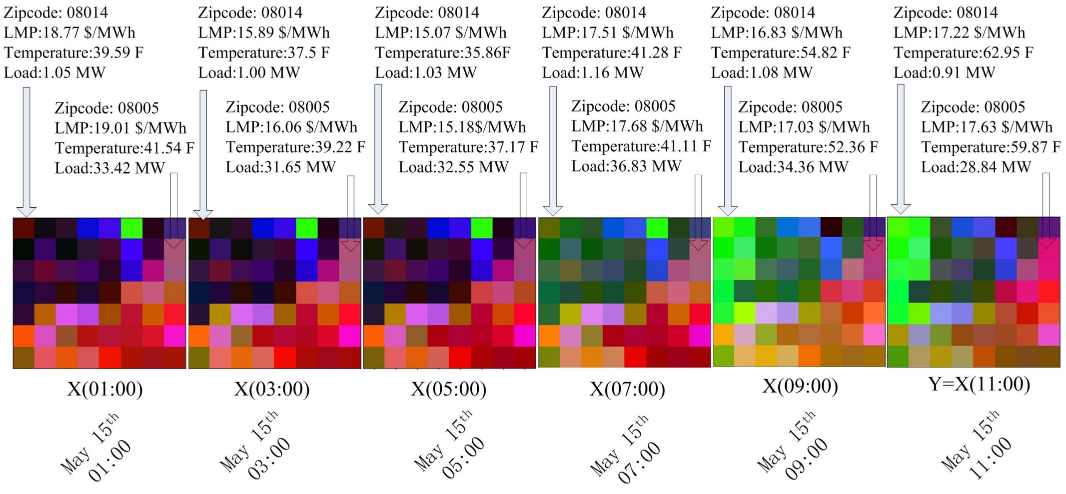


Figure 3.11: Detailed Information Stored in Colored Digital Market Data Video of AECO Price Zone in PJM Market.

Although in the above discussion, the market data pixel $x_{i,j}^{norm}(t)$ is represented using three types of market data (such as RTLMP, real power demand, and temperature in Fig. 3.10 and Fig. 3.11), this concept of market data pixel can be easily extended to representing l different types of market data, where l is a positive integer. In chapter 4.3, a feature selection process is introduced to identify the l types of market data which contribute the most to the LMP prediction problem.

3.3 Formulation of The RTLMP Prediction Problem

The normalized historical market data across the system for T hours are organized into a normalized historical market data tensor $\mathcal{X}^{norm} = \{X^{1-norm}, \cdots, X^{t-norm}, \cdots, X^{T-norm}\}$. The spatio-temporal correlations among values in the RTLMP channel are determined by the unknown PTDF matrix $\mathbf{T}$ and uncertainties of other market data channels. Thus, a predictor should be employed to learn the implicit relationships stored in this historical market data tensor $\mathcal{X}^{norm}$ and estimate a reasonable 2D array $\hat{X}^{(T+1)-norm}$ for next time $T+1$. The estimated 2D array $\hat{X}^{(T+1)-norm}$ contains predicted RTLMPs for the next time step.

The RTLMP prediction problem is formulated as a 2D array prediction problem. Given a historical market data tensor as input, the objective of the desired predictor is to generate a future 2D array $\hat{X}^{(T+1)-norm}$, such that the predicted $\hat{X}^{(T+1)-norm}$ is close to the ground truth $X^{(T+1)-norm}$. Meanwhile, spatio-temporal correlations in $\hat{\mathcal{X}}^{norm} = \{X^{1-norm}, \cdots, X^{t-norm}, \cdots, X^{T-norm}, \hat{X}^{(T+1)-norm}\}$ is similar to those in the ground truth $\mathcal{X}^{norm} = \{X^{1-norm}, \cdots, X^{t-norm}, \cdots, X^{T-norm}, X^{(T+1)-norm}\}$. This is achieved by maximizing the conditional probability $p(\hat{X}^{(T+1)-norm}|\mathcal{X}^{norm})$ using the proposed predictors.

3.3.1 One-Hour Ahead LMP Prediction

As shown in Fig. 3.12, the input of one-hour ahead LMP predictor is a tensor consisting of several hours' historical market data ($\mathcal{X} = \{X^1, X^2, X^3, X^4, X^5, X^6, X^7\}$). The output of the predictor is the forecast of the next hour's frame ($\hat{X}^8$). The objec-

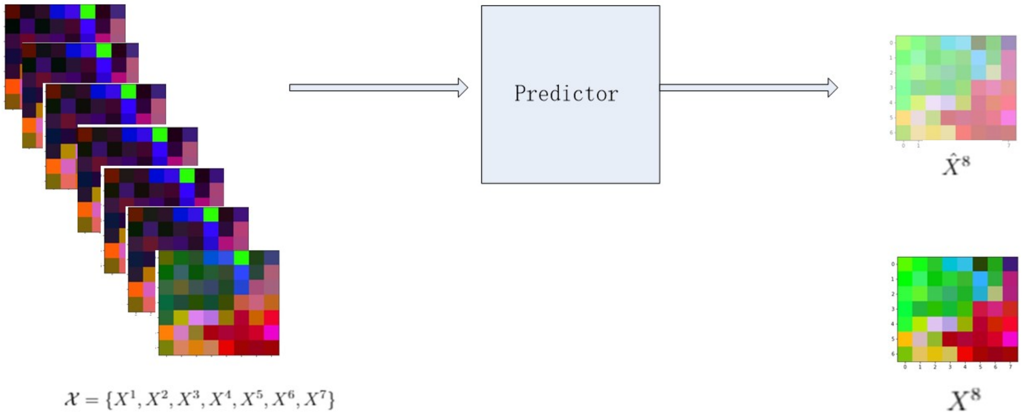


Figure 3.12: Example of One-hour Ahead LMP Prediction.

tive is to maximize the conditional probability $p(\hat{X}^{T+1}|\mathcal{X})$ (i.e. $p(\hat{X}^{8}|\mathcal{X})$ in Fig. 3.12) or minimize the distance between forecast $\hat{X}^{T+1}$ and the ground truth X^{T+1} (i.e. the distance between $\hat{X}^{8}$ and X^{8} in Fig. 3.12).

3.3.2 Day-Ahead RTLMP Prediction

The input of the one-day ahead LMP predictor is a tensor consisting of several days' historical market data ($\mathcal{X}_{Day} = \{X^1, \cdots, X^d, \cdots, X^D\}$). The output of the predictor is the forecast of the next day's frame $\hat{X}^{D+1}$. The objective is to maximize the conditional probability $p(\hat{X}^{D+1}|\mathcal{X}_{Day})$ or minimize the distance between forecast $\hat{X}^{D+1}$ and the ground truth X^{D+1}.

Chapter 4

GENERATIVE ADVERSARIAL NETWORK-BASED PREDICTION

A deep convolutional generative adversarial network (GAN) model is proposed to solve the LMP prediction problem. The convolutional GAN model is trained using multiple loss functions that can capture spatio-temporal correlations among system-wide historical market data.

4.1 Convolutional Generative Adversarial Network

4.1.1 Convolutional Neural Network

The convolutional neural network (CNN) is one of the main techniques designed for image recognition, image classification, and image generation. The general structure of CNN includes convolutional layers, activation units, pooling layers, and fully connected layers as shown in Fig. 4.1.

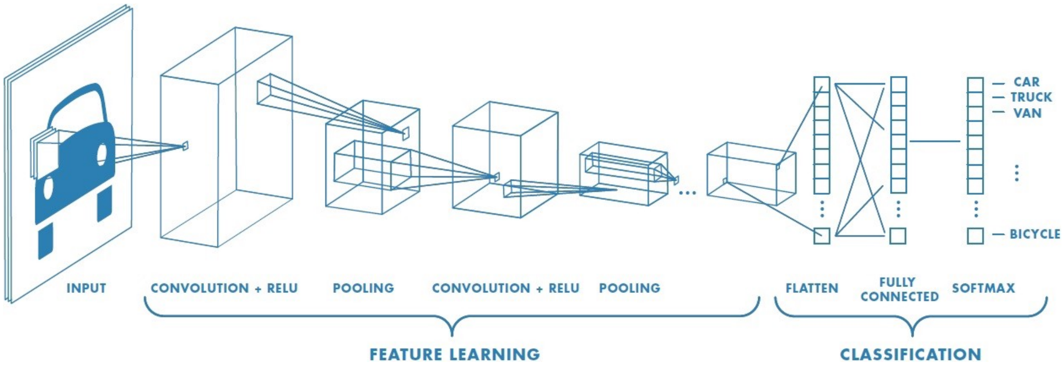
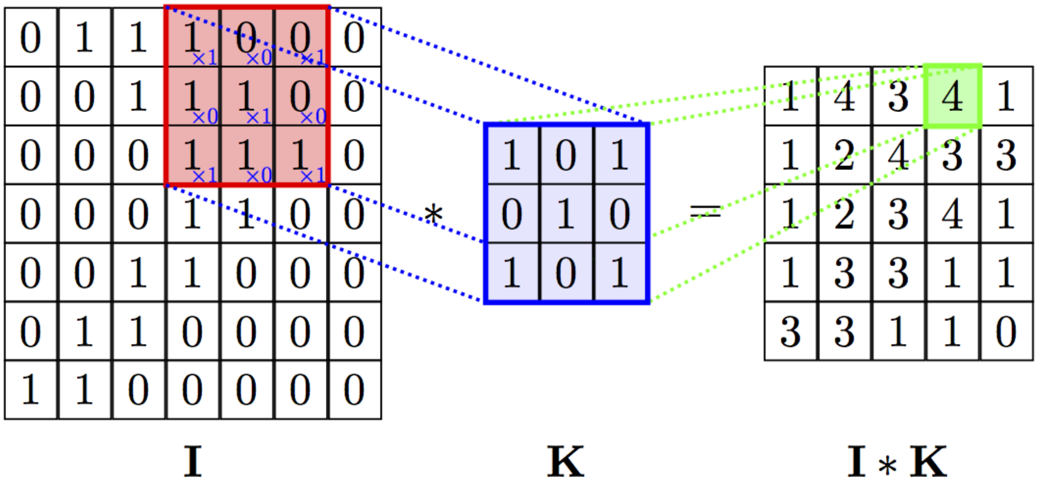


Figure 4.1: General CNN Structure [6].

Convolutional layers serve as the primary components within CNNs. These layers are composed of learnable filters, also known as kernels. Each filter is spatially compact, spanning only a small width and height, yet it covers the entire depth of the input volume. By employing convolution operation, the correlation between pixels is preserved, allowing the filters to learn image features effectively. A convolution operation between an image and a filter(kernel) can be mathematically expressed as:

$$C[m][n] = \sum_{u} \sum_{v} (I[m+u][n+v] \cdot K[u][v]) \tag{4.1}$$

where I is the image, K is the kernel of size $u \times v$. The result C of a convolution is a feature map of the original image I filtered by kernel K, as the example shown in Fig. 4.2. Intuitively, the value in a feature map represents how much the kernel



0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	1	1	0	0	0	0
1	1	0	0	0	0	0

1	0	1
0	1	0
1	0	1

1	4	3	4	1
1	2	4	3	3
1	2	3	4	1
1	3	3	1	1
3	3	1	1	0

Figure 4.2: Example of Convolution.

captures the correlation in the patch of the original image.

In addition to convolutions, pooling operations make up another important building block in CNNs. Pooling operations reduce the size of feature maps by using some function to summarize subregions, such as taking the average or the maximum value. Padding ways and stride size also impact the size of feature maps. A guide to convolution arithmetic can be found in [68].

In CNN, the convolutional layers are followed by fully connected layers, as shown in Fig. 4.3.

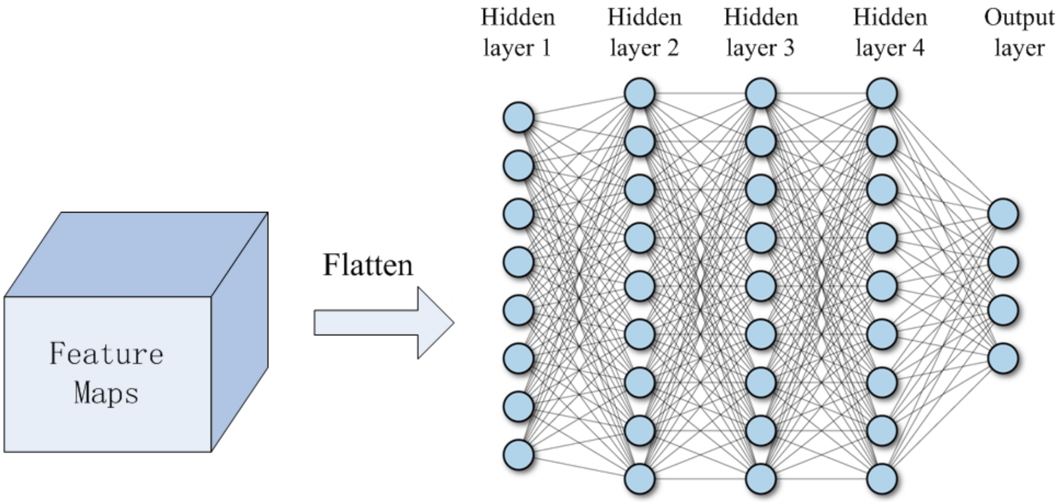


Figure 4.3: Fully Connected Layers.

Each node in Fig. 4.3 represents a neuron unit:

$$a_h^t = \sum_{h'=1}^{I} w_{h'h} b_{h'}^{t-1} \tag{4.2}$$

$$b_h^t = \theta_h(a_h^t) \tag{4.3}$$

where, $w_{h'h}$ are the weights between neuron h' in layer $t-1$ and neuron h in layer t; I is the total number of neurons in layer $t-1$; a_h^t is the weighted sum from last layer; θ_h is the nonlinear activation function for neuron h; b_h^t is the output of neuron h in layer t and input to layer $t+1$. The neuron mathematical representation is shown in Fig. 4.4. Similar to convolutional layers, fully connected layers are also capable of abstracting features from their input.

4.1.2 Generative Model G

We propose a convolutional generative model as a solution to capture correlations among pixels in input images using filters (kernels). This model, referred to as the generator (G) and illustrated in Fig. 4.5, is designed to generate future images based on a sequence of historical frames. Through training, the convolutional generative

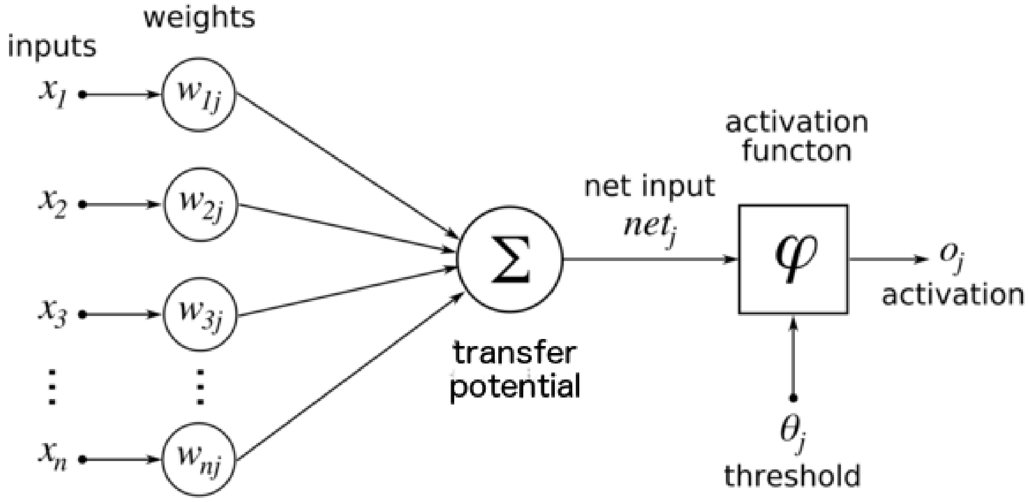


Figure 4.4: Neuron Mathematical Representation.

model learns to predict the succeeding frame. This convolutional generative model

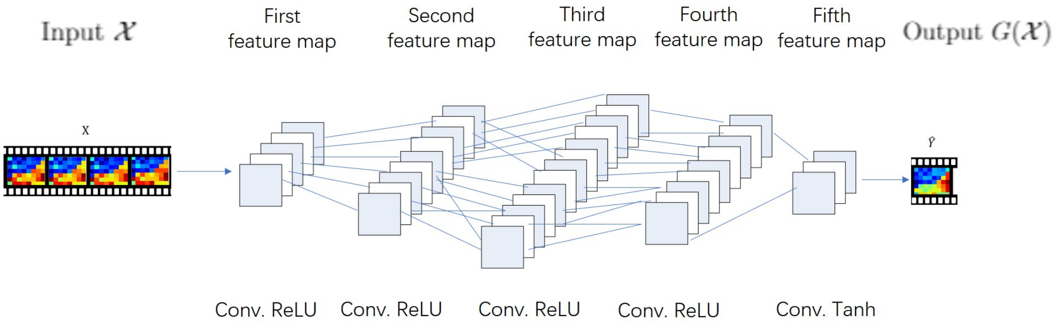


Figure 4.5: A Basic next Frame Generative Model G

G consists of several convolution layers followed by nonlinear activation functions. Rectified linear units (ReLU) are employed after each convolution layer, except for the last one. Since pixel values are expected to fall within the range of [-1, 1], the output of the final convolution layer is passed through the hyperbolic tangent function (Tanh) to ensure the desired range of pixel values.

As shown in Fig. 4.5, the generator G's input $\mathcal{X} = \{X^1, \cdots, X^n\}$ denotes the 3D tensor (video) consisting series of matrices (images) with normalized historical market data (at different price nodes/locations) at n consecutive time instants, $\mathcal{X}\subset\mathcal{X}^{norm}$; $\hat{Y}$ denotes the generated 2D array (image) with forecasted normalized LMPs for time $n+1$, i.e., $\hat{Y}$ is the forecast of Y. Y denotes the ground truth 2D array (image) with normalized historical LMPs at time $n+1$ to forecast, i.e., $Y = X^{n+1}\in\mathcal{X}^{norm}$; Unlike traditional CNN, the output of G has to be of the same resolution as the input. The pooling layers, which reduce the size of images, are removed, and transposed convolution layers with padding are accepted to keep the image resolution.

Intuitively, the generator G takes historical market data in a 3D tensor $\mathcal{X} = \{X^1, \cdots, X^n\}$ as inputs and is trained to generate the next frame $\hat{Y} = G(\mathcal{X})$ of the given tensor, such that a distance between the forecast $\hat{Y}$ and ground truth Y is minimized. The distance between two 2D arrays can be measured by p-norm distance ℓ_p:

$$\mathcal{L}_p(\mathcal{X}, Y) = \ell_p(G(\mathcal{X}), Y) = \ell_p(\hat{Y}, Y) = \|G(\mathcal{X}) - Y\|_p^p \tag{4.4}$$

where $\|\cdot\|_p$ denotes the entry-wise p-norm of a particular matrix. When $p = 2$, the above loss function measures the Euclidean distance between $\hat{Y}$ and Y; when $p = 1$, the loss function measures the Manhattan distance between $\hat{Y}$ and Y.

During the training process, this loss function forces the generator G to generate the next-hour market data 2D array $\hat{Y}$ that is close to the corresponding ground-truth Y by minimizing the p-norm distance between Y and $\hat{Y}$.

However, predicting RTLMPs only with such normal CNN-based G has two drawbacks:

1. Using p-norm loss may lead to loss of spatial correlations. In actual energy markets, congestion only happens in limited lines. Because the PTDF matrix $\mathbf{T}$ in (1.3) is very sparse, most price nodes have pretty similar LMPs, which are close to the marginal energy price λ. In the training process, minimizing

p-norm loss over the historical dataset may result in constant RTLMPs for all price nodes and loss of spatial correlations.

2. Using p-norm loss may lead to blurry predictions [69]. According to equation (1.3), it is evident that LMPs can vary even when considering the same PTDF matrix and identical supply-demand scenarios. LMPs can vary depending on the bidding variables provided by generator owners. Consequently, the probability distributions of LMPs are influenced by the corresponding probability distributions of bidding strategies. Suppose generator owners have two equally likely strategies s_1 and s_2, with probability $p(s_1) = p(s_2)$, the corresponding LMP at one price node has two values x_{s_1} and x_{s_2} with same probabilities $p(x_{s_1}) = p(x_{s_2})$ over the dataset. During the training process, minimizing p-norm loss over the dataset will lead to the blurry prediction $x_{avg} = (x_{s_1} + x_{s_2})/2$, even if the probability $p(x_{avg})$ is very low in the historical dataset.

To address the issue of blurriness, we utilize an additional adversarial model to maximize the conditional probability $P(\hat{Y}|\mathcal{X})$. This enables us to capture the spatio-temporal correlations among various frames within a single 3D tensor. Furthermore, we incorporate several supplementary losses into the overall multi-loss function to enhance the learning process and account for these correlations.

4.1.3 Discriminative Model D

To introduce an adversarial loss, we incorporate a discriminative model, referred to as the discriminator (denoted as D). The purpose of the discriminator is to maximize the conditional probability $P(\hat{Y}|\mathcal{X})$. The architecture of the discriminator, as depicted in Fig. 4.6, is based on a CNN design.

The discriminator D is incorporated in the adversarial training process to avoid

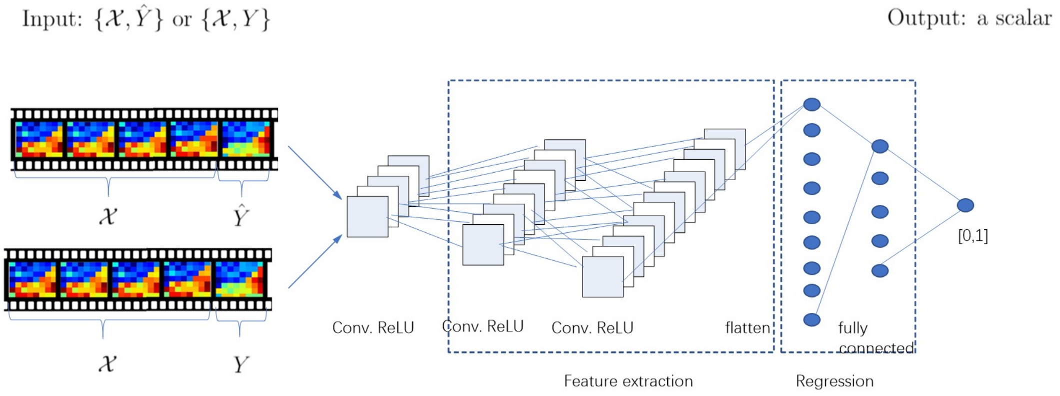


Figure 4.6: A Basic Discriminative Model D.

blurry predictions by learning the spatio-temporal correlations among historical market data. $\{\mathcal{X}, Y\}$ is a constructed tensor including $\mathcal{X}$ and Y, representing the ground truth tensor; $\{\mathcal{X}, \hat{Y}\}$ is a constructed tensor including $\mathcal{X}$ and $\hat{Y}$, representing the generated tensor (with the generated 2D array $\hat{Y}$ concatenated after the ground-truth historical tensor $\mathcal{X}$). D takes $\{\mathcal{X}, Y\}$ or $\{\mathcal{X}, \hat{Y}\}$ as input, and outputs a scalar $D(\{\mathcal{X}, \cdot\}) \in [0, 1]$ to indicate the probability of the input $\{\mathcal{X}, \cdot\}$ being the ground-truth. During training, the target of $D(\{\mathcal{X}, Y\})$ is class 1; the target of $D(\{\mathcal{X}, \hat{Y}\})$ is class 0.

To gain a better understanding of the concept, we can consider the same example we discussed earlier. For example, given a historical tensor $\mathcal{X}$ in the dataset, the probability distribution for the next frame has two equally likely values Y_{s1} and Y_{s2} ($P(Y = Y_{s1}|\mathcal{X}) = P(Y = Y_{s2}|\mathcal{X})$). As discussed above, G trained with p-norm loss will result in the blurry prediction $\hat{Y} = Y_{avg} = (Y_{s1} + Y_{s2})/2$. However, as the input of D, the concatenated tensor $\{\mathcal{X}, Y_{avg}\}$ is not an equally likely tensor to $\{\mathcal{X}, Y_{s1}\}$ and $\{\mathcal{X}, Y_{s2}\}$ in the same dataset. Therefore, D can easily discriminate them and penalize the blurry predictions. The only tensors the discriminator D will not be able to classify as fake are $\{\mathcal{X}, Y_{s1}\}$ and $\{\mathcal{X}, Y_{s2}\}$. Thus the objective of D is to minimizing a loss function to classify the input $\{\mathcal{X}, Y\}$ into class 1 (i.e., Y is classified as the ground-truth 2D array) and the input $\{\mathcal{X}, \hat{Y}\} = \{\mathcal{X}, G(\mathcal{X})\}$ into class 0 (i.e., $\hat{Y} = G(\mathcal{X})$ is classified as the generated fake 2D array). Then the output of D can be utilized to penalize G with blurry predictions. We will talk about the loss functions and adversarial training procedure in the following sections.

4.1.4 Multi-Loss Function

With the objective discussed above, we define the loss function for discriminator D and generator G respectively.

Loss function for training D

The adversary loss function for discriminator D is defined as:

$$\mathcal{L}_{adv}^{D}(\mathcal{X}, Y) = \mathcal{L}_{bce}(D(\{\mathcal{X}, Y\}), 1) + \mathcal{L}_{bce}(D(\{\mathcal{X}, G(\mathcal{X})\}), 0) \tag{4.5}$$

where $\mathcal{L}_{bce}$ is the following binary cross-entropy:

$$\mathcal{L}_{bce}(k, s) = -[slog(k) + (1 - s)log(1 - k)] \tag{4.6}$$

where, $k \in [0, 1]$ and $s \in \{0, 1\}$. The binary cross-entropy measures the probability distribution distance between the discriminator outputs $K = D(\{\mathcal{X}, \cdot\})$ and the associated labels S ($S_i = 1$ and $S_i = 0$ for real and generated 2D arrays, respectively). These S labels are attached in the training process based on the learning algorithm, instead of given with the training dataset, therefore the proposed approach is unsupervised.

With this adversary loss function, the discriminator D forces its output scalar $D(\{\mathcal{X}, Y\})$ to 1, and $D(\{\mathcal{X}, \hat{Y}\}) = D(\{\mathcal{X}, G(X)\})$ to 0. Note the difference between input tensor $\{\mathcal{X}, Y\}$ and input tensor $\{\mathcal{X}, \hat{Y}\}$ is only the last 2D array in the third dimension which represents the temporal correlation of market data between the last timestamp and previous timestamp. In this way, the discriminator D takes advantage of learning the temporal correlations in the historical ground-truth tensors and discriminates the ground-truth 2D array Y from the fake 2D array $\hat{Y} = G(\mathcal{X})$ generated by the generator G, given the input historical market data 3D tensor $\mathcal{X} = \{X^1, \cdots, X^n\}$.

Loss function for training G

To fully learn the spatio-temporal correlations among historical market data tensor and reduce blurry predictions, a multi-loss function is adopted for the generator G. This multi-loss function includes four terms from [70; 71] to quantify the distances between $\hat{Y}$ and Y from various perspectives, as while as penalize blurry predictions:

$$\mathcal{L}^G(\mathcal{X}, Y) = \lambda_{adv}\mathcal{L}^G_{adv}(\mathcal{X}, Y) + \lambda_{\ell_p}\mathcal{L}_p(\mathcal{X}, Y) + \lambda_{gdl}\mathcal{L}_{gdl}(\mathcal{X}, Y) + \lambda_{dcl}\mathcal{L}_{dcl}(\mathcal{X}, Y) \quad (4.7)$$

where $\mathcal{L}^G(\mathcal{X}, Y)$ denotes the overall weighted loss function for training G; $\mathcal{L}^G_{adv}(\mathcal{X}, Y)$, $\mathcal{L}_p(\mathcal{X}, Y)$, $\mathcal{L}_{gdl}(\mathcal{X}, Y)$, and $\mathcal{L}_{dcl}(\mathcal{X}, Y)$ denote the four individual loss function terms (introduced separately in the following sections); λ_{adv}, λ_{ℓ_p}, λ_{gdl}, and λ_{dcl} denote the hyperparameters (weights) for adjusting the tradeoffs among these loss terms.

1. **The p-norm Loss Function $\mathcal{L}_p(\mathcal{X}, Y)$:** The first loss function is the p-norm loss function measuring the entry-wise distance between the generated and ground-truth market data images $\hat{Y} = G(\mathcal{X})$ and Y:

   $$\mathcal{L}_p(\mathcal{X}, Y) = \ell_p(G(\mathcal{X}), Y) = \|G(X) - Y\|_p^p \quad (4.8)$$

   As shown in the example in Fig. 4.7, the p-norm loss function accumulates the Euclidean distance or Manhattan distance between each pair of pixels in the same location of $G(\mathcal{X})$ and Y for the whole images. During the training

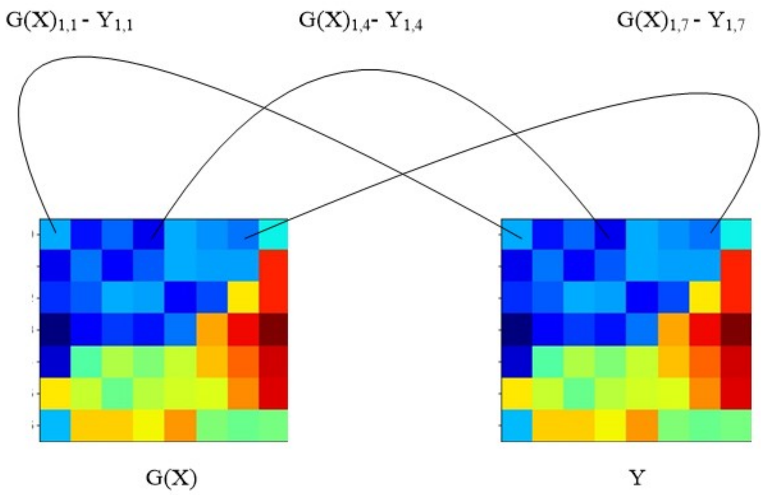


Figure 4.7: A Representation of $\mathcal{L}_p(\mathcal{X}, Y)$.

process, this loss function forces the generator G to generate the future market data 2D array $\hat{Y}$ that is close to the corresponding ground-truth 2D array Y by minimizing the difference between Y and $\hat{Y}$.

2. **The Adversarial Loss Function $\mathcal{L}_{adv}^{G}(\mathcal{X}, Y)$:** According to the above discussion, we know the p-norm loss will cause blurry predictions which can be easily identified by D. Therefore, we can take advantage of the outputs from D to penalize these blurry predictions during the training process. The following adversarial loss function $\mathcal{L}_{adv}^{G}(\mathcal{X}, Y)$ is adopted to penalize the G by learning the temporal correlations among historical market data:

$$\mathcal{L}_{adv}^{G}(\mathcal{X}, Y) = \mathcal{L}_{bce}(D(\{\mathcal{X}, G(\mathcal{X})\}), 1) \tag{4.9}$$

During the training process, the adversarial loss function forces the generator G to generate the future market data 2D array $\hat{Y}$ that is temporally coherent with the input historical market data tensor $\mathcal{X}$, such that the generated tensor $\{\mathcal{X}, \hat{Y}\}$ is realistic enough to confuse the discriminator D. This is achieved by minimizing $\mathcal{L}_{bce}(D(\{\mathcal{X}, \hat{Y}\}), 1)$, which measures the probability distribution distance between the discriminator output for the generated market data tensor and the label for real market data tensor. Penalized by this loss factor, G tends to generate 2D arrays that are temporally coherent with the historical 2D arrays in the ground-truth tensor.

3. **The Gradient Difference Loss Function $\mathcal{L}_{gdl}(\mathcal{X}, Y)$:** The following loss function is adopted to further capture the spatial correlations among historical market data:

$$\begin{aligned} \mathcal{L}_{gdl}(\mathcal{X}, Y) &= \mathcal{L}_{gdl}(\hat{Y}, Y) \\ &= \sum_{i,j} ||Y_{i,j} - Y_{i-1,j}| - |\hat{Y}_{i,j} - \hat{Y}_{i-1,j}||^{\alpha} \\ &+ ||Y_{i,j-1} - Y_{i,j}| - |\hat{Y}_{i,j-1} - \hat{Y}_{i,j}||^{\alpha} \end{aligned} \tag{4.10}$$

where $\alpha \geq 1, \alpha \in \mathbb{Z}$; $Y_{i,j}$ and $\hat{Y}_{i,j}$ denote $(i,j)^{th}$ elements in the ground-truth and generated market data 2D array Y and $\hat{Y} = G(\mathcal{X})$ as shown in Fig.4.8, respectively. During the training process, this loss function forces the generator G

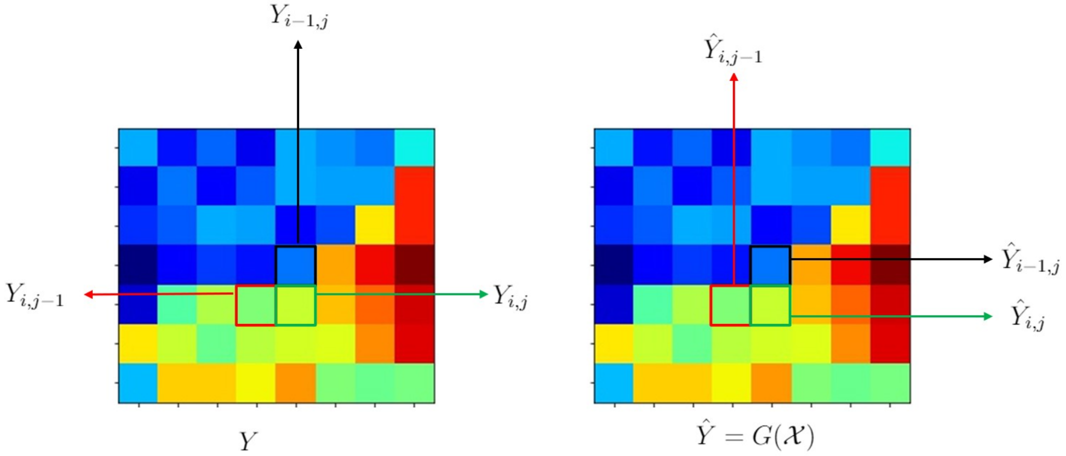


Figure 4.8: A Representation of $\mathcal{L}_{gdl}(\mathcal{X}, Y)$.

to generate the future market data 2D array $\hat{Y}$ that has similar locational gradient difference information compared to the ground-truth 2D array Y. Since the locational gradient difference information captures the spatial variations of historical market data, this minimization ensures the generated market data fully captures the spatial correlations among system-wide historical market data.

4. **The Direction Changing Loss Function** $\mathcal{L}_{dcl}(\mathcal{X}, Y)$ Unlike classical video prediction problems, we would like to correctly predict whether LMPs will increase or decrease in the next market clearing interval. The following loss function is introduced to capture the changing directions of market data at specific locations over time:

$$\mathcal{L}_{dcl}(\mathcal{X}, Y) = \sum_{i,j} |sgn(\hat{Y}_{i,j} - x^t_{i,j}) - sgn(Y_{i,j} - x^t_{i,j})| \tag{4.11}$$

where $Y_{i,j}$, $\hat{Y}_{i,j}$ and $x^t_{i,j}$ are the $(i, j)^{th}$ elements in the ground-truth 2D array, forecasted 2D array and the last frame of given historical tensor as shown in Fig. 4.9, respectively $sgn(\cdot)$ is the sign function:

$$sgn(x) = \begin{cases} -1 & if\ x \leq 0 \\ 0 & if\ x = 0 \\ 1 & if\ x \geq 0 \end{cases} \tag{4.12}$$

During the training process, this loss function forces the generated market data tensor $\{\mathcal{X}, \hat{Y}\}$ to correctly follow the pixel changing directions in the ground-truth tensor $\{X, Y\}$ by penalizing incorrect prediction of market data trend over time.

4.1.5 Adversarial Training

The GAN-based RTLMP prediction model is trained through the adversarial training procedure in Fig. 4.10. Algorithm 1 summarizes the training algorithm.

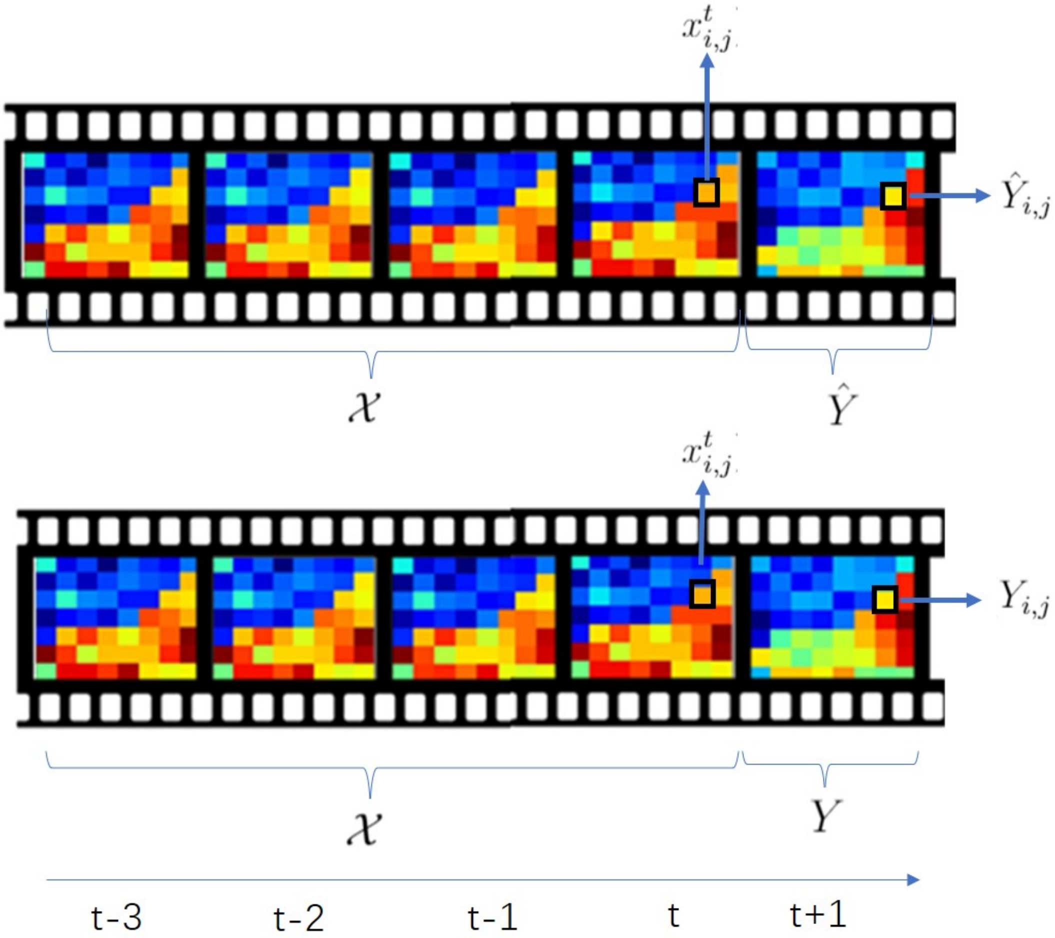


Figure 4.9: A Representation of $\mathcal{L}_{dcl}(\mathcal{X}, Y)$.

The generator G and discriminator D are trained simultaneously, with their model weights, W_G and W_D, updated iteratively. The stochastic gradient descent (SGD) minimization is adopted to obtain optimal model weights. In each training iteration, a new batch of M training data samples (i.e., M historical market data videos) are obtained for updating W_G and W_D. More precisely, in discriminator iteration, let $(\mathcal{X}, Y)$ be a sample of M videos with their next frame from the dataset. Then, we perform one SGD update of D while keeping the weights of G fixed. Then, let $(\mathcal{X}, Y)$ be a different data sample of M videos with their next frame from the dataset in the following generator iteration. One SGD update of G is performed while keeping the weights of D fixed. Upon convergence, the generator G is trained to generate $\hat{Y}$ as realistic as possible, such that the discriminator D cannot confidently classify $\{X, \hat{Y}\}$ into 0 as a generated video.

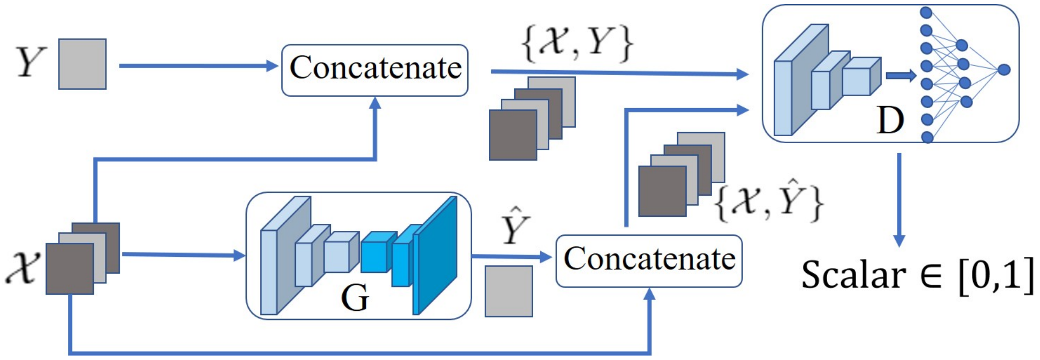


Figure 4.10: The Training Procedure of the GAN-based LMP Prediction Model.

During training, the appropriate values of 4 hyperparameters λ_{adv}, λ_{ℓ_p}, λ_{gdl}, and λ_{dcl} are selected by the following 2-step tuning procedure:

1. *Rough selection*: The hyperparameters λ_{adv} and λ_{ℓ_p} are first determined in the rough selection step. These two hyperparameters are associated with the basic loss functions of classic GANs. Therefore, without appropriate selection of λ_{adv} and λ_{ℓ_p}, the GAN training may encounter convergence issues. In this step, λ_{ℓ_p} is set at 1; λ_{gdl}, and λ_{dcl} are set at 0; the range of λ_{adv} is set between 0 and 0.5 with a step size of 0.05. The learning rates are set as $\{0.00005, 0.0005, 0.005, 0.05, 0.5\}$. The hyperparameter λ_{adv} is selected by grid search [72].

2. *Fine-tuning*: After the rough selection step, λ_{adv}, λ_{ℓ_p} and the learning rates are fixed. Then, λ_{gdl}, and λ_{dcl} are determined by fine tuning.

The initial guess of these parameters is from [69]. The final hyperparameters are

Algorithm 1 Training generative adversarial networks for RTLMP prediction

```
Require: set the learning rates $\rho_D$ and $\rho_G$, loss hyperparameters $\lambda_{adv}$, $\lambda_{\ell_p}$, $\lambda_{gdl}$, $\lambda_{dcl}$,
  and minibatch size M
Require: initial discriminative model weights $W_D$ and generative model weights $W_G$

  while not converged do
    Update the discriminator D:
    Get a batch of M data samples from the training dataset, $(\mathcal{X}, Y) = (\mathcal{X}^{(1)}, Y^{(1)}), \cdots, (\mathcal{X}^{(M)}, Y^{(M)})$
    Do one SGD update step
    $W_D = W_D - \rho_D \sum_{i=1}^{M} \frac{\partial \mathcal{L}_{adv}^{D}(\mathcal{X}^{(i)}, Y^{(i)})}{\partial W_D}$
    Update the generator G:
    Get a new batch of M data samples from the training dataset, $(\mathcal{X}, Y) = (\mathcal{X}^{(1)}, Y^{(1)}), \cdots, (\mathcal{X}^{(M)}, Y^{(M)})$
    Do one SGD update step
    $W_G = W_G - \rho_G \sum_{i=1}^{M} (\lambda_{adv} \frac{\partial \mathcal{L}_{adv}^{G}(\mathcal{X}^{(i)}, Y^{(i)})}{\partial W_G} + \lambda_{\ell_p} \frac{\partial \mathcal{L}_{\ell_p}(\mathcal{X}^{(i)}, Y^{(i)})}{\partial W_G} + \lambda_{gdl} \frac{\partial \mathcal{L}_{gdl}(\mathcal{X}^{(i)}, Y^{(i)})}{\partial W_G} + \lambda_{dcl} \frac{\partial \mathcal{L}_{dcl}(\mathcal{X}^{(i)}, Y^{(i)})}{\partial W_G})$
  end while
```

determined through the above tuning procedure to obtain the best performance.

4.2 Calibration

The GAN model described in Chapter 4.1 is trained using year-long historical market data and applied to perform RTLMP predictions step by step for the following year. Due to fuel price fluctuation, load growth, and generation/transmission systems upgrade, the market data statistics may vary year by year. This may cause deviations between the ground truth and predicted RTLMPs, as the generator G

is trained using market data from previous years. Fig. 4.11 shows the comparison among hourly RTLMP histograms from 2012 to 2018 in ISO-New England market. It is clear that although RTLMPs obtained during different years for the same market follow similar probability distributions, there exist discrepancies between RTLMP distributions during different years. These statistical discrepancies may decrease the RTLMP prediction accuracy if the prediction model is trained only using historical data. The result in Chapter 4.4 shows that RTLMPs prediction generated directly from G trained by historical data will have biases from the ground-truth RTLMPs. To compensate for these deviations and further improve the forecasting accuracy, two calibration methods can be applied to the outputs of G: (1) moving average calibration (MA); (2) autoregressive moving average calibration (ARMA).

4.2.1 Moving Average Calibration

In the cases studied in Chapter 4.4, the generative model G is trained to generate the next frame $\hat{Y}$, given a historical market data tensor consisting of 4 consecutive frames $\mathcal{X} = \{X^{t-3}, X^{t-2}, X^{t-1}, X^{t}\}$. The following moving average (MA) approach is used to calibrate the prediction:

$$\widetilde{Y}^{t+1} = \widehat{Y}^{t+1} - \frac{\sum_{i=t-3}^{t}(\widehat{Y}^{i} - Y^{i})}{4} \tag{4.13}$$

where Y^i denotes the 2D array with ground-truth RTLMPs at time i; $\widetilde{Y}^{t+1}$ denotes the calibrated 2D array with forecasted RTLMPs at time $t+1$; $\widehat{Y}^{i}$ denotes the 2D array generated by the generator G at time i. Using (4.13), we calibrate the forecasted RTLMPs for the next interval with the average difference between the RTLMPs generated by G and the ground-truth RTLMPs over the past four intervals, which are exactly the length of input tensor of G. The framework for RTLMPs prediction is shown in Fig. 4.12.

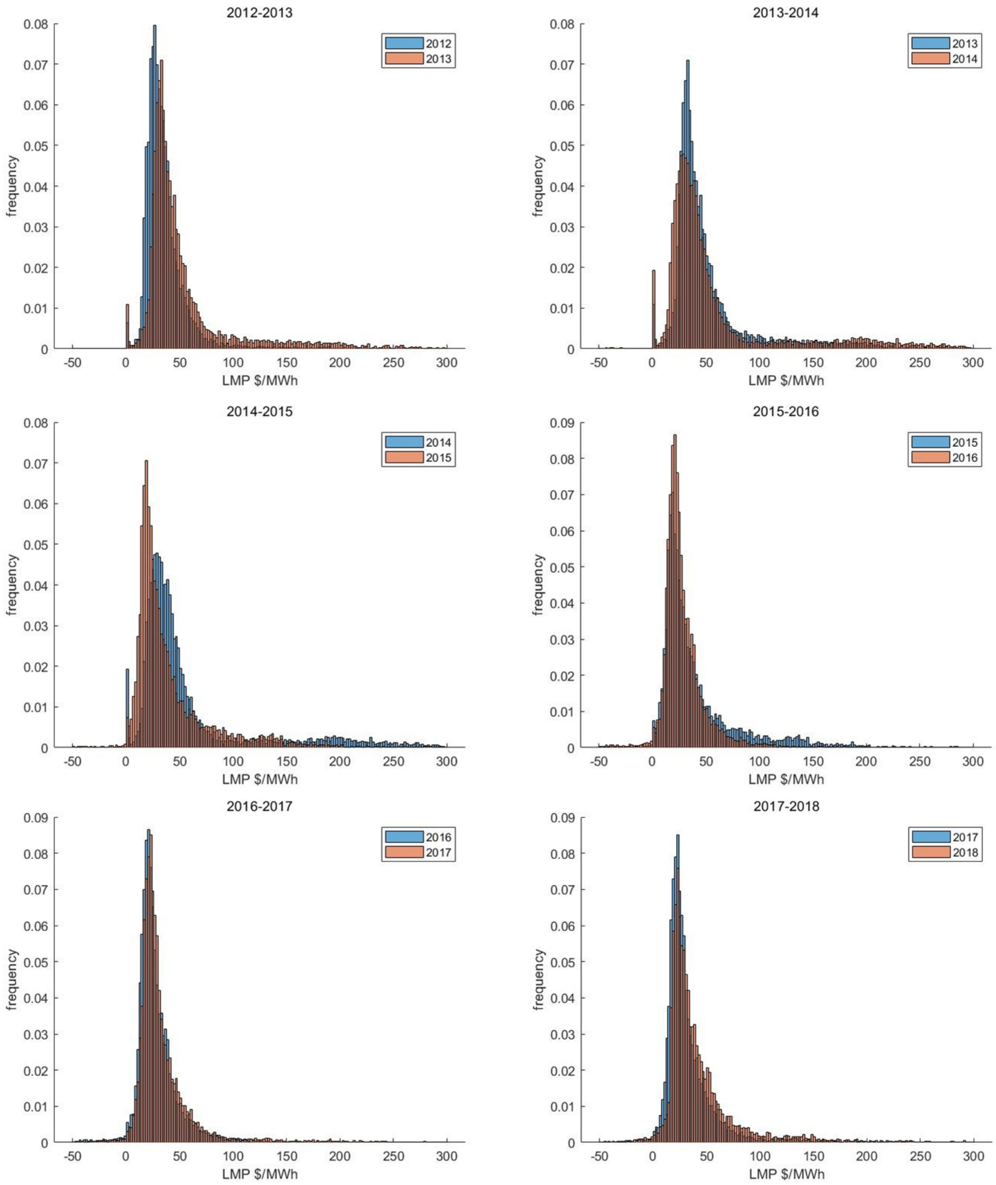


Figure 4.11: Comparison among Hourly RTLMP Distribution from 2012 to 2018 in ISO-NE.

4.2.2 Autoregressive Moving Average Calibration

MA calibration accounts for short-term dependency over the past several time steps. A series of deviations between the ground-truth 2D arrays and the generated

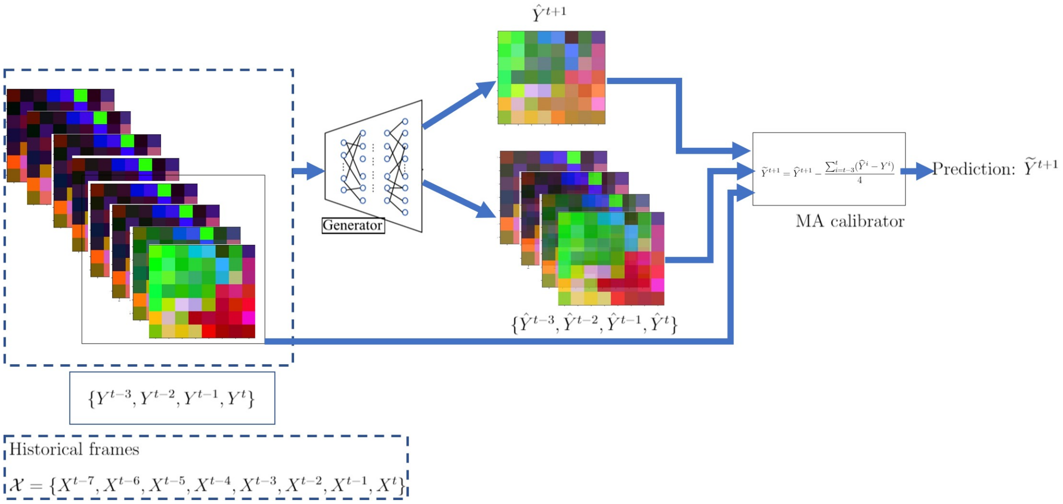


Figure 4.12: The Framework for RTLMPs Prediction with MA Calibration.

2D arrays can be denoted as:

$$\Delta Y^i = Y^i - \hat{Y}^i \tag{4.14}$$

where Y^i denotes the ground-truth 2D array at time i; $\hat{Y}^i$ denotes the generated RTLMP 2D array from generator G for time i; ΔY^i denotes the deviation between the generated 2D array and ground-truth 2D array. For instance, we use the market data of 2017 in ISO-NE to train the GAN model, then predict the RTLMPs of 2018 hour by hour using the trained G. The stationarity of $\Delta Y = \{Y^{i-t}, Y^{i-t+1}, \cdots, Y^{i-1}, Y^i\}$ of 2018 for ISO-NE prediction model is demonstrated by the sample autocorrelation function (ACF) shown in Fig. 4.13. Because ACF shown in Fig. 4.13 decays quickly, the sequence of ΔY is appropriate to fit in an ARMA model.

For better prediction accuracy, the RTLMPs generated by GAN are calibrated by estimating their deviations from the ground truth:

$$\widetilde{y}(i+1) = \hat{y}(i+1) + \Delta\hat{y}(i+1) \tag{4.15}$$

where

$$\Delta\hat{y}(i+1) = y(i+1) - \hat{y}(i+1) + e(i+1) \tag{4.16}$$

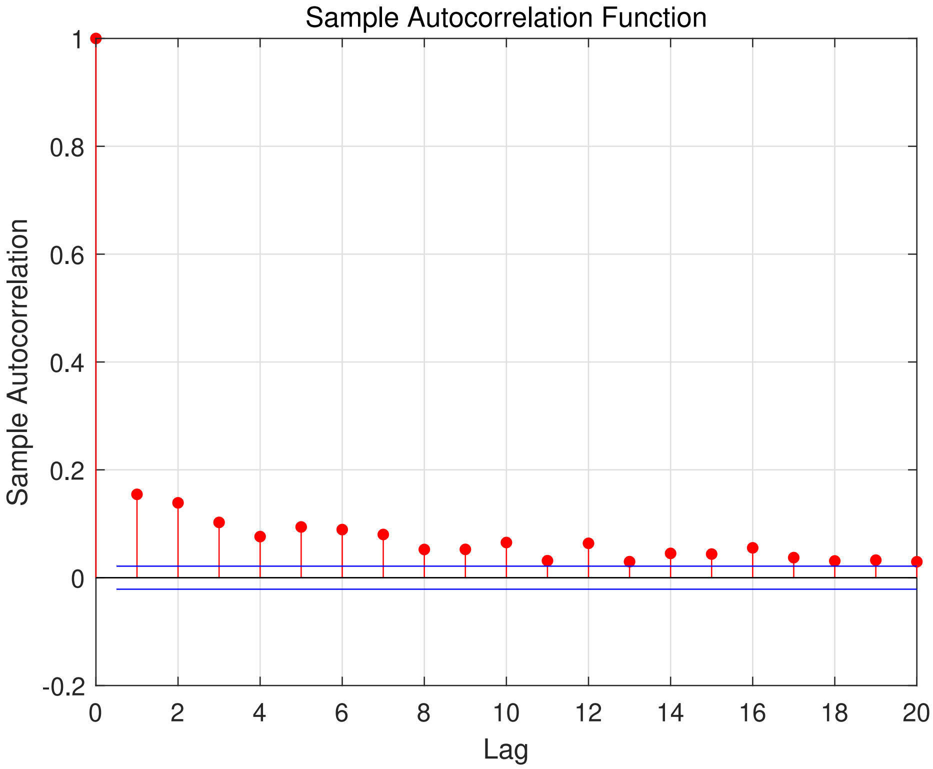


Figure 4.13: ACF Plots of ΔY for ISO-NE RTLMP Prediction in 2018.

where $y(i+1)$, $\hat{y}(i+1)$, and $\widetilde{y}(i+1)$ denote the ground-truth RTLMP, the RTLMP generated by G, and the RTLMP after calibration at time $i+1$ for a particular price node, respectively; $\Delta\hat{y}(i+1)$ denotes the estimated difference between $y(i+1)$ and $\hat{y}(i+1)$; $e(i+1)$ denotes the estimation error.

The ARMA model below is applied to estimate $\Delta\hat{y}(i+1)$:

$$\begin{aligned}\Delta\hat{y}(i+1) =&\mu + \sum_{k=1}^{p} \phi_k \Delta\hat{y}(i-k+1) \\ &+ \sum_{k=1}^{q} \theta_k \varepsilon(i-k+1) + \varepsilon(i+1)\end{aligned} \tag{4.17}$$

where μ denotes the expectation of $\Delta\hat{y}(i+1)$, ϕ_k and θ_k are the autoregressive (AR) and MA parameters of the ARMA model, respectively; $\varepsilon(i)$ represents the white noise error terms at time i; p and q denote the orders of the AR and MA terms of the ARMA model, respectively. Appropriate values of p, q, μ, ϕ_k, θ_k, and the variance of the white noise series $\varepsilon(i)$ are identified using long-term historical data [73; 74; 75].

The overall framework of the proposed design for RTLMP prediction is shown in Fig. 4.14. The proposed RTLMP prediction framework includes the following steps:

1. Data normalization and organization. Historical public data streams are organized into a time series of 2D arrays (i.e., the 3D tensor).
2. RTLMP predictions are generated from the GAN-based predictor G for the next hour and several past hours.
3. RTLMP predictions for the next hour are calibrated through an ARMA model using generated RTLMPs and the ground truth of historical data.

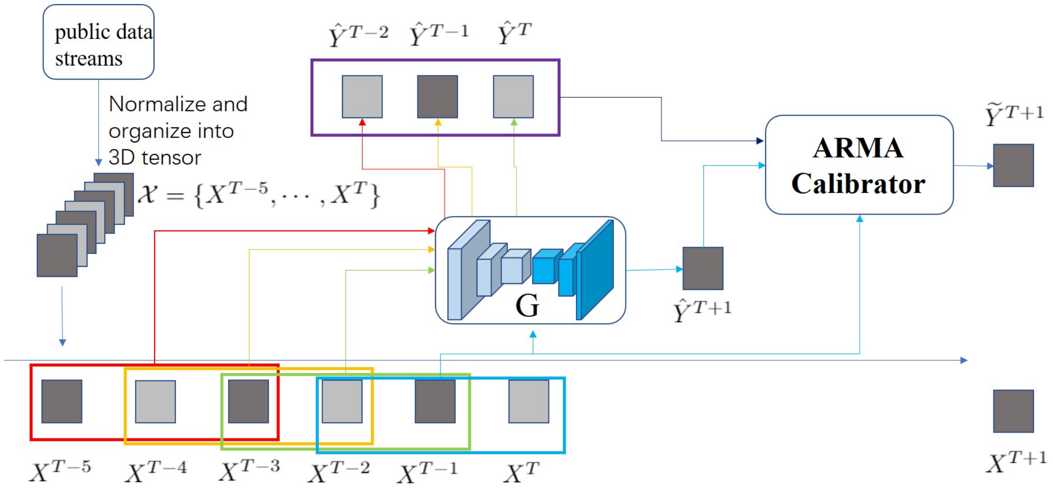


Figure 4.14: The Framework of GAN-based Approach with ARMA Calibrator for LMP Prediction.

4.3 Feature Selection

Based on the data structure defined previously, each market data pixel in the GAN prediction model may contain M different types of market data points (after normalization). This section identifies the following four types of market data that are highly related to the RTLMP prediction problem and are publicly available in many US electricity markets.

4.3.1 RTLMP

To learn the spatio-temporal correlations among historical RTLMPs, the proposed prediction model is trained using historical RTLMPs. As discussed in Chapter 4.2, the historical RTLMPs in the same market obtained from different years follow similar probability distributions with discrepancies. They also follow similar daily, weekly, and seasonal characteristic which is the temporal correlations to learn by the prediction model. Besides temporal correlations, RTLMPs collected from different price nodes in the same market tend to have similar spatial correlations over different years. For example, Fig. 4.15 demonstrates RTLMPs' spatial correlation coefficients among 9 price zones in ISO-NE (8 price zones and one system price) over three years. It

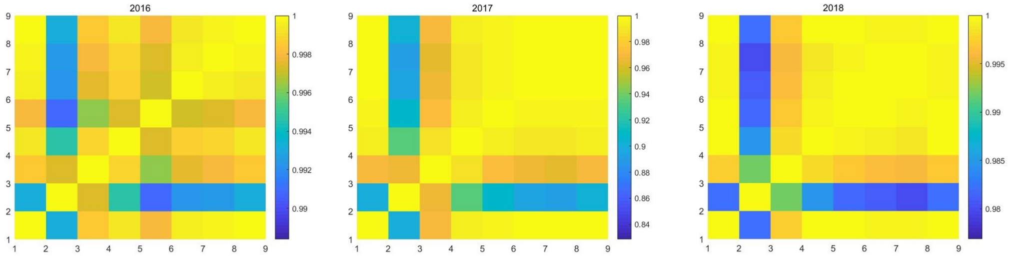


Figure 4.15: The Spatial Correlation Coefficients Matrix Heatmap Generated using 2016 RTLMPs (Left), 2017 RTLMPs (Central), and 2017 RTLMPs Rright) in ISO-NE.

is clear that the RTLMPs follow similar spatial characteristics over the years, which makes it possible the GAN model can learn spatial correlations from the historical training dataset.

4.3.2 DALMP

RTLMPs tend to be strongly correlated with DALMPs, since generators committed in the day-ahead market may affect the total generation capacity and overall energy price in the real-time market. As an example, the correlation coefficient between DALMPs and RTLMPs in ISO-NE is 66.38% in 2018 [3]. Fig. 4.16 shows clearly that RTLMPs and DALMPs are highly correlated with each other, they follow similar daily, weekly and seasonal characteristics. Therefore, historical DALMPs are adopted as training inputs for the GAN model.

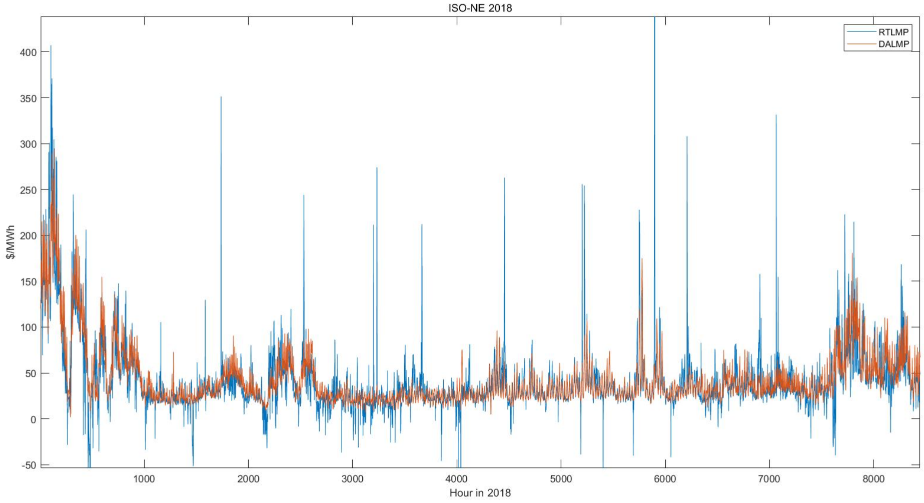


Figure 4.16: The System RTLMPs and DALMPs of ISO-NE in 2018.

4.3.3 Demand

System demand patterns and uncertainties could significantly affect RTLMPs and DALMPs. As an example, the correlation coefficient between zonal DALMPs and Demands in ISO-NE is 51.35% in 2018 [3]. Since locational demand data is public in many electricity markets, these historical demands are included as training inputs for our GAN model.

4.3.4 Generation

Generation price and quantity offers play critical roles in the LMP formation process. Therefore, historical generation offer data contains important information for predicting RTLMPs. However, in most electricity markets, system-wide generation offer details are published with 3 months of delays, and therefore cannot be fully utilized by market participants for their price prediction problem. To resolve this issue, we adopt historical hourly generation mix data that is publicly available in some markets as additional inputs in training our GAN model. Table 4.1 shows the correlation coefficients calculated between SPP's historical generation mix (i.e., percentage generation by fuel type) data and historical RTLMP data [4]. It is clear that the percentage generations of certain generation types are highly correlated with RTLMPs in SPP. Therefore, historical hourly generation mix data is adopted in training the RTLMP prediction model for SPP. Some markets, such as PJM shown in Table. 3.1, provide locational wind and solar generation data. These renewable energy generations impact LMP significantly since they typically serve as marginal units in the electricity markets. Thus, these public locational renewable energy generation data can be taken as additional inputs.

Table 4.1: Correlations Between Generation Mix and the RTLMPs in SPP

Generation Type	**Correlation Coefficient**
Coal market	17.83%
Coal self	22.72%
Diesel fuel oil	-5.26%
Hydro	19.73%
Wind	-37.93%
Natural gas	36.25%
Nuclear	-29.34%
Solar	17.20%
Waste disposal	-14.83%

4.4 Case Studies

The GAN-based prediction method is tested using historical market data from ISO-NE, SPP, and MISO. Two GAN models are built using Tensorflow 2.0 [76]: (1) one-hour ahead GAN prediction model (Model 1); (2) one-day ahead GAN prediction model (Model 2). Model 1 is tested to predict hourly RTLMPs one-hour ahead for ISO-NE and SPP, and hourly DALMPs one-hour ahead for MISO. Model 2 is tested in SPP to predict hourly RTLMPs one-day ahead.

4.4.1 Neural Network Architecture and Configuration

Both the generative and discriminative models are deep convolutional neural networks without any pooling/subsampling layers. In the generative models, all the transpose convolutional (Conv2DTranspose) layers are followed by the batch normalization layers and ReLU units. The outputs of the generative model are normalized by a hyperbolic tangent (Tanh) function. In the discriminative model, except for the output layer, all the convolutional (Conv2D) layers and fully-connected (Dense) layers are followed by the batch normalization layers, Leaky-ReLU units, and dropout layers.

All the convolutional (Conv2D) and transpose convolutional (Conv2DTranspose) layers in model 1 are with kernel size of 3×3 and stride size of 1×1. All the convolutional (Conv2D) and transpose convolutional (Conv2DTranspose) layers in model 2 are with kernel size of 6×6 and stride size of 6×6. The transpose convolutional (Conv2DTranspose) layers in the generative models are padded, and the convolutional (Conv2D) layers in discriminative models are not padded. In the discriminative models, the dropout rates are set to 0.3, the small gradients are set to 0.2 when Leaky-ReLU is not active. All the neural networks are trained using a standard SGD optimizer with a minibatch size of 4, i.e., $M = 4$ in Algorithm 1. The learning rates ρ_G and ρ_D for model 1 are set to 0.0005, without decay and momentum. The learning rates ρ_G and ρ_D for model 2 are set to 0.000005 and 0.00001, without decay and momentum. The loss functions for model 1 in (4.7)-(4.11) are implemented with the following parameteres: $\lambda_{adv} = \lambda_{dcl} = 0.2$ (in (4.7)), $\lambda_{\ell_p} = \lambda_{gdl} = 1$ (in (4.7)), $p = 2$ (in (4.8)), and $\alpha = 1$ (in (4.11)). The loss functions for model 2 in (4.7)-(4.11) are implemented with the following parameteres: $\lambda_{adv} = 0.2$ (in (4.7)), $\lambda_{\ell_p} = \lambda_{gdl} = 1$, $\lambda_{dcl} = 0$ (in (4.7)), $p = 2$ (in (4.8)), and $\alpha = 1$ (in (4.11)).

Summaries of the neural network architecture are listed in Table 4.2 and Table 4.3.

4.4.2 Test Case Description

The proposed approach is applied to predict zonal-level RTLMPs in ISO-NE [3], SPP [4] and MISO [5]. For both markets, historical market data from nine price zones are organized into 3×3 hourly market data 2D array X and 3D tensor $\mathcal{X}$. For

Table 4.2: Neural Network Architecture Details of Model 1

Model 1	Generator G (Layer Type, Feature Map)	Discriminator D (Layer Type, Feature Map)
Input	$3 \times 3 \times 14$	$3 \times 3 \times 5$
Layer 1	Conv2DTranspose, 64	Conv2D, 64
Layer 2	Concatenate, 1216	Concatenate, 320
Layer 3	Conv2DTranspose, 1024	Dense, 1024
Layer 4	Conv2DTranspose, 512	Dense, 512
Layer 5	Conv2DTranspose, 64	Dense, 256
Output	$3 \times 3 \times 1$	scalar$\in [0, 1]$

Table 4.3: Neural Network Architecture Details of Model 2

Model 2	Generator G (Layer Type, Feature Map)	Discriminator D (Layer Type, Feature Map)
Input	$12 \times 18 \times 14$	$12 \times 18 \times 5$
Layer 1	Conv2DTranspose, 64	Conv2D, 64
Layer 2	Concatenate, 1216	Concatenate, 320
Layer 3	Conv2DTranspose, 256	Dense, 1024
Layer 4	Conv2DTranspose, 128	Dense, 512
Layer 5	Conv2DTranspose, 64	Dense, 256
Output	$12 \times 18 \times 1$	scalar$\in [0, 1]$

the SPP market, the same data are also organized into 12×18 daily market data 2D array X^d and 3D tensor $\mathcal{X}_{Day}$. The test case data is described as follows.

Case 1

The training data set contains three types of hourly ISO-NE market data (zonal RTLMPs, DALMPs, demands) from 1/1/2017 to 12/31/2017. The trained model is tested by predicting ISO-NE RTLMPs hour by hour in 2018.

Case 2

The training data set contains three types of hourly ISO-NE market data (zonal RTLMPs, DALMPs, demands) from 1/1/2016 to 12/31/2017. The trained model is tested by predicting ISO-NE RTLMPs hour by hour in 2018.

Case 3

The training data set contains four types of hourly SPP market data (zonal RTLMPs, DALMPs, demands, and generation mix data) from 6/1/2016 to 7/30/2017. The model is tested by predicting SPP RTLMPs hour by hour in the following four periods: 7/31/2017-8/13/2017, 8/21/2017-9/3/2017, 9/18/2017-10/1/2017, and 10/2/2017-10/15/2017.

Case 4

The training data set contains four types of hourly SPP market data (zonal RTLMPs, DALMPs, demands, and generation mix data) from 6/1/2016 to 7/30/2017. The model is tested by predicting SPP RTLMPs day by day in the following four periods: 7/31/2017-8/13/2017, 8/21/2017-9/3/2017, 9/18/2017-10/1/2017, and 10/2/2017-10/15/2017.

Case 5

The training data set [77] contains three types of hourly MISO market data (zonal DALMPs, demands, and wind generation) from January 2012 to November 2014. The trained model is tested by predicting MISO DALMPs hour by hour in an hour-ahead manner on several representative days selected in [77], for NPPD, Arkansas, Louisiana, and Texas hubs.

Case 6

The training data set [77] contains three types of hourly MISO market data (zonal DALMPs, demands, and wind generation) from January 2012 to November 2014. The trained model is tested by predicting MISO DALMPs hour by hour in a day-ahead manner on several representative days selected in [77], for the Indiana hub.

The test cases are summarized in Table. 4.4.

4.4.3 Results and Performance Analysis

Table 4.5 lists the results of hyperparameter selection in Case 5. After this two-step hyperparameter tuning procedure, appropriate hyperparameters with the lowest MAPE are adopted in the GAN-based price prediction model. Table 4.5 also indicates the GAN-based predictor with multiple loss functions and adversarial training (with non-zero values for all the 4 hyperparameters) has much higher prediction accuracy (lower MAPE) compared to the CNN-based predictor without adversarial training (with $\lambda_{adv} = \lambda_{gdl} = \lambda_{dcl} = 0$) or with only a subset of the 4 loss functions (with zero values for certain hyperparameters).

The RTLMP prediction accuracy is quantified by the mean absolute percentage error (MAPE). Table 4.6 shows the RTLMP prediction accuracy of our proposed approach in Case 1-2. For the ISO-NE test Case 1, the annual MAPEs in 2018 is

Table 4.4: Test Case Summary

Case #	**Market**	**Time horizon**	**Training data**	**Test data**	**Resolution**
1	ISO-NE	one-hour ahead	1/1/2017-12/31/2017	1/1/2018-12/31/2018	3×3
2	ISO-NE	one-hour ahead	1/1/2016-12/31/2017	1/1/2018-12/31/2018	3×3
3	SPP	one-hour ahead	6/1/2016-7/30/2017	7/31/2017-8/13/2017, 8/21/2017-9/3/2017, 9/18/2017-10/1/2017, 10/2/2017-10/15/2017	3×3
4	SPP	one-day ahead	6/1/2016-7/30/2017	7/31/2017-8/13/2017, 8/21/2017-9/3/2017, 9/18/2017-10/1/2017, 10/2/2017-10/15/2017	12×18
5	MISO	one-hour ahead	1/1/2016-8/30/2014	9/1/2014, 10/12/2014	3×3
6	MISO	one-day ahead	1/1/2016-8/30/2014	9/1/2014, 10/12/2014	12×18

around 15% for all nine price zones. With the same GAN model but a longer training dataset in Case 2, the annual MAPEs in 2018 is around 11% for all the nine price zones. Since the proposed price prediction approach is a data-driven approach, a longer training dataset can apparently improve prediction accuracy.

There are other studies predicting RTLMPs in ISO-NE [78; 79]. In [78], a weekly

Table 4.5: MAPEs (%) for Different Hyperparameter Selections in Case 5

	λ_{ℓ_p}	λ_{adv}	λ_{gdl}	λ_{dcl}	MAPE
Rough	1	0	0	0	6.51
Selection	1	0.2	0	0	5.14
Fine	1	0.2	1	0	4.73
Tuning	1	0.2	0	0.2	4.67
	1	0.2	1	0.2	4.38

MAPE of 10.87% is achieved by predicting daily average on-peak-hour prices in ISO-NE. Since the daily average on-peak-hour price is the daily averaged price from hour 8 to hour 23, this averaged price is expected to have much smoother behavior and fewer spikes compared to the hourly RTLMPs in our test case. When tested on raw hourly RTLMPs for an entire year without averaging/smoothing over the testing data, our approach can achieve reasonable accuracy compared to [78]. When tested on raw hourly RTLMPs obtained for the same week as [78] at a different year, our approach achieves a weekly MAPE of 9.08%, which outperforms the work in [78] using averaged prices. In [79], an average MAPE of 10.81% is achieved by predicting ISO-NE prices at four test weeks in March, June, September, and December. However, the testing data in [79] is generated through simulations. The simulation rules assume all possible explicit price-responsive behavior is known, which is not practical in real-world price prediction. Our proposed approach, when tested using year-long actual market data, has very similar MAPEs compared to [79].

In [80; 81] similar hour-by-hour price prediction approaches are tested using public data from Japan market (with a price prediction MAPE of 14.28%) and the Spanish market (with a price prediction MAPE of 15.83%), respectively. Our approach has

Table 4.6: Zonal RTLMP Prediction Accuracy in Case 1-2

Case #	**Zone**	**MAPE**
Case 1	VT	15.80%
	HN	15.91%
	ME	16.55%
	WCMA	15.82%
	System	15.72%
	NEMA	15.69%
	CT	15.91%
	RI	15.78%
	SEMA	15.72%
Case 2	VT	11.03%
	HN	11.25%
	ME	11.82%
	WCMA	10.99%
	System	11.06%
	NEMA	11.05%
	CT	11.04%
	RI	11.01%
	SEMA	11.05%

much lower MAPEs compared to these works.

Fig. 4.17 shows the ground truth and predicted RTLMPs for VT price zone in ISO-NE during the entire year of 2018. It is clear our predicted RTLMPs closely follow the overall trends of the ground-truth RTLMPs, and successfully capture most price spikes in the testing window.

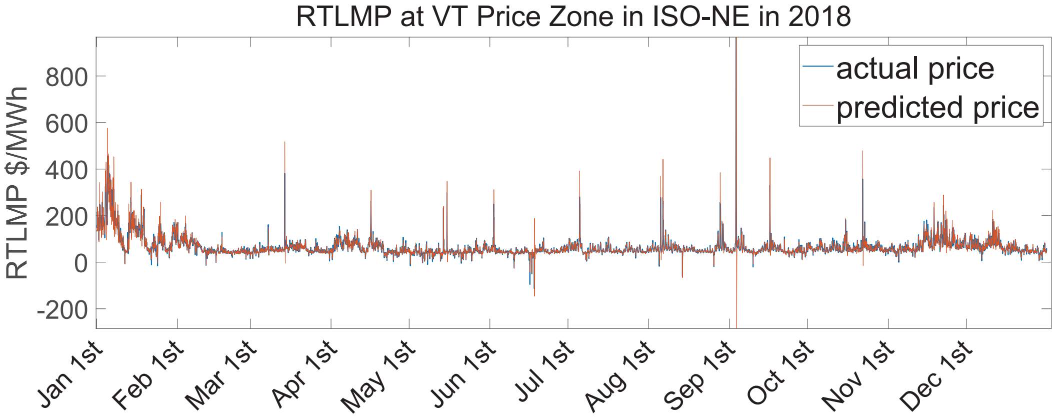


Figure 4.17: Ground-truth and Predicted RTLMPs for VT Price Zone in Case 1.

In Case 3, the same GAN model is trained and tested using SPP data. Table 4.7 shows the RTLMP prediction accuracy in Case 3. It is worth mentioning that the prediction accuracies in Case 3 (SPP) are lower than those in Case 1 and Case 2 (ISO-NE). This is because SPP has a larger market territory and more market participants compared to ISO-NE. These facts lead to more complex and harder-to-predict market dynamics in SPP.

In Case 4, a GAN model with a larger resolution is trained by the same SPP data as in Case 3 and tested by predicting the next day's hourly RTLMPs. Table 4.8 shows the RTLMP forecast accuracy of the proposed method in Case 3, Case 4, and the MAPEs obtained by implementing of two benchmark approaches in [1] using the same testing data at the south hub and north hub in SPP real-time market. ALG+$\hat{M}$, Genscape and Case 4 forecast hourly RTLMPs with one-day ahead; Case 3 forecasts hourly RTLMPs with one-hour ahead. ALG+$\hat{M}$ [1], Case 3 and Case 4 predict LMPs using only publicly available data; Genscape incorporates richer and proprietary data which are confidential to market participants. We observe that the proposed approach has comparable performance to the state-of-art industry benchmark Genscape, using only

Table 4.7: Zonal RTLMP Prediction Accuracy in Case 3

Case #	**Zone**	**MAPE**
Case 3	Nhub	18.90%
	NPPD	19.26%
	OPPD	18.22%
	LES	18.59%
	CSWS	17.40%
	GRDA	17.64%
	WFEC	17.97%
	OKGE	17.68%
	SHub	16.94%

limited data. Our GAN-based forecast model performs remarkably better than the existing approach in [1], using the same public market data. The ALG+$\hat{M}$ method proposed in [1] does not utilize DALMP data in RTLMP forecasting. For a fair comparison, an additional GAN-based model is trained without DALMPs in Case 4. These approaches are tested using the identical testing dataset from SPP's SHub and North Hub (NHub) zones.

Fig. 4.18 compares the ground-truth RTLMPs, the forecasted RTLMPs without moving average calibration, and the forecasted RTLMPs with moving average calibration at the south hub over the whole testing period in Case 3. Fig. 4.18 shows that without the moving average calibration, the RTLMPs forecasted by the proposed GAN model can successfully capture the temporal correlations in the ground-truth RTLMPs. However, there exists a constant bias between the ground-truth RTLMPs and the RTLMPs forecasted without calibration. After applying the moving aver-

Table 4.8: RTLMP Prediction Accuracy in Case 3-4 and [1]

Approach	**MAPE (%) for SHub Price Zone**	**MAPE (%) for NHub Price Zone**
ALG+$\hat{M}$[1]	25.4	36.9
Genscape[2]	21.7	28.2
Case 3	16.9	18.9
Case 4	22.1	23.8
Case 4 without DALMPs	23.5	25.4

[1] The proposed method with the best performance in [1]

[2] State of art baseline prediction from Genscape [1]

age calibration, this bias is corrected and the forecasting accuracy is improved. It's clear that the forecasted RTLMPs after calibration closely follow the overall trends of the ground-truth RTLMPs, agree with the temporal characteristics, and successfully capture several price spikes.

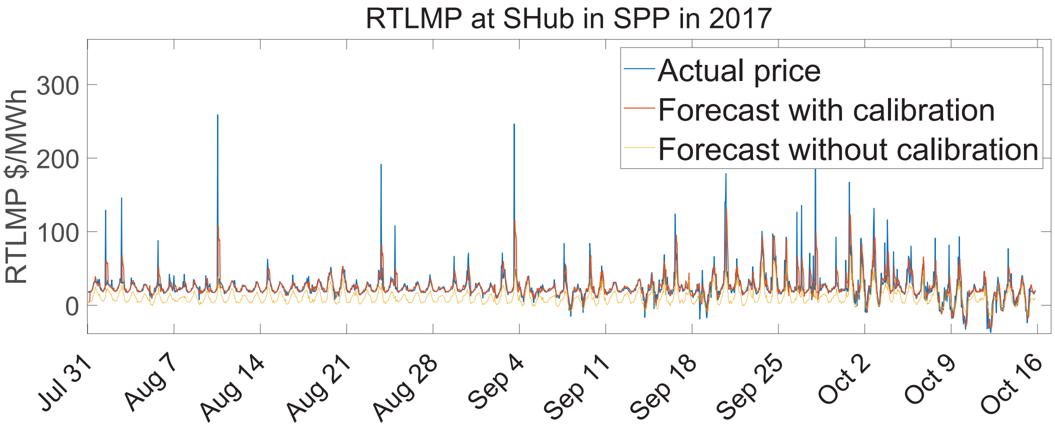


Figure 4.18: Ground-truth and Forecasted RTLMPs (with and Without Calibration) at South Hub in Case 3.

In the testing window of 8/21/2017-9/3/2017 of Case 3, as shown in Fig. 4.19, the proposed GAN model successfully captures price spikes on Aug 21, Aug 29, Aug 30, and Sep 2 at different locations. Spatial characteristic is well learned by the proposed model. On Aug 21, the local spike appearing only at CSWS is precisely captured by our approach. This local spike prediction could be utilized by local market participants to develop their risk-averse trading strategies and save significant losses caused by price spikes.

To further demonstrate the capability of learning spatio-temporal correlations, Fig. 4.20 compares the spatial correlations obtained using predicted RTLMPs with those obtained using ground-truth RTLMPs in Case 3. Each 9×9 heatmap matrix in Fig. 4.20 visualizes the spatial correlation coefficients among 9 price zones in SPP. It is clear our RTLMP prediction model successfully captures the spatial correlations across SPP market.

In Case 5, the MAPEs of the representative days for Louisiana and Texas hubs are computed and compared with benchmarking models in Table 4.9, in which the SDA and RS-SDA prediction models are proposed by [77]; The ETs prediction model is proposed by [64]. In [77], SDA and RS-SDA are tested on the same case using the same data. The results of SDA and RS-SDA directly come from [77]. The ETs model includes three individual ET regressors for LMP components prediction and one meta-regressor for the final summation of LMPs. In Case 5, the regressors of the ETs model are trained using historical energy prices, congestion prices, and loss prices. Only the final LMP prediction results of the ETs model are listed in Table 4.9.

Table 4.9 demonstrates the proposed GAN prediction model outperforms all the other benchmarking models for hour-ahead DALMP prediction in all the testing scenarios.

Fig. 4.21 shows the testing results for hour-ahead DALMP prediction in Case 5

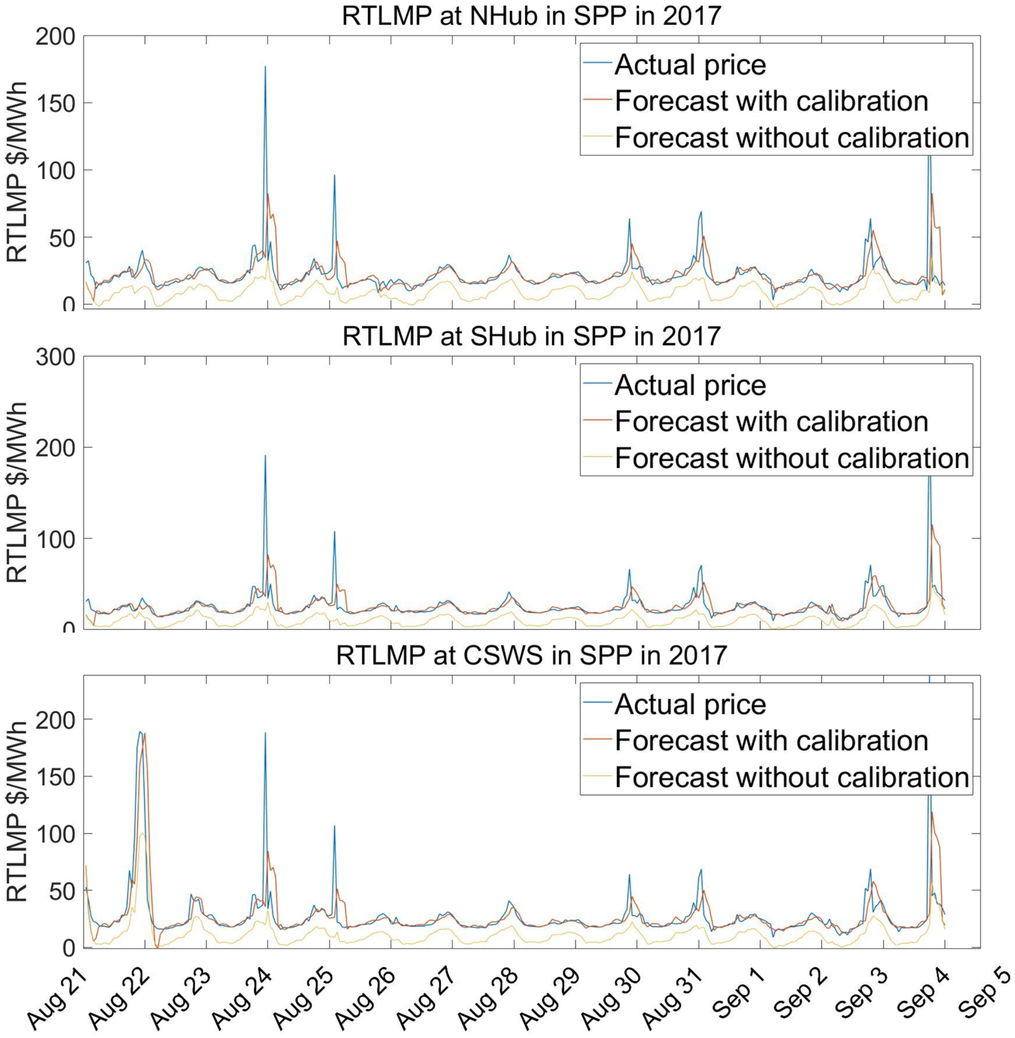


Figure 4.19: Ground-truth and Forecasted RTLMPs at NHub, SHub, and CSWS in Case 3.

on one representative day. It is observed the GAN-based model successfully captures different daily price characteristics for different locations.

To further evaluate the effectiveness of the gradient difference loss function $\mathcal{L}_{gdl}(\cdot)$ and the direction changing loss function $\mathcal{L}_{dcl}(\cdot)$ for penalizing incorrect spatial correlation predictions, two additional metrics, the spatial prediction accuracy (SPA) and the spatial mean absolute percentage error (SMAPE), are defined below to quantify

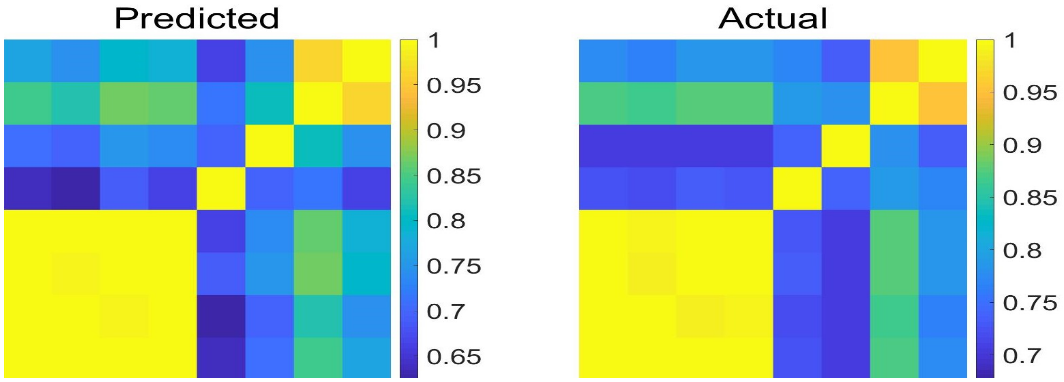


Figure 4.20: The Spatial Correlation Coefficients Matrix Heatmap Generated using Predicted RTLMPs (Left) and Ground-truth RTLMPs (Right) in Case 3.

Table 4.9: MAPE (%) for Hour-Ahead Forecasting in Case 5

Model	Naive(A)	Naive(B)	NN	MARS	SVM	LASSO
Texas	31.45	5.4	6.19	6.54	7.34	7.88
Louisiana	27.34	9.15	6.19	6.28	6.14	7.70
Model	ARMA	ETs	SDA	RS-SDA	GAN	
Texas	7.02	6.43	5.42	5.16	4.83	
Louisiana	6.84	6.09	4.66	4.51	4.13	

the spatial correlation prediction accuracy/error at a price node n.

$$SPA(n) = 1 - \frac{\sum_{i=1}^{N} |sgn(Y_n - Y_i) - sgn(\hat{Y}_n - \hat{Y}_i)|/2}{N-1} \tag{4.18}$$

$$SMAPE(n) = \frac{\sum_{i=1}^{N} |(Y_n - Y_i) - (\hat{Y}_n - \hat{Y}_i)|}{\sum_{i=1}^{N} |(Y_n - Y_i)|} \tag{4.19}$$

where N represents the total number of price nodes; Y_i represents the ground truth LMP at node i $(i \neq n)$; $\hat{Y}_i$ represents the predicted LMP at node i $(i \neq n)$; $sgn(\cdot)$ is the sign function in (4.12). $SPA(n)$ measures the average prediction accuracy for the price increasing or decreasing directions between node n and all other nodes.

$SMAPE(n)$ measures the average prediction error for the price differences between node n and all other nodes. In this report, higher SPA and lower SMAPE indicate better prediction accuracy of spatial correlations.

Table 4.10 compares the SPA and SMAPE metrics for the Texas hub in Case 5, which indicates the GAN-based model outperforms other models in forecasting spatial correlations among DALMPs for the Texas hub.

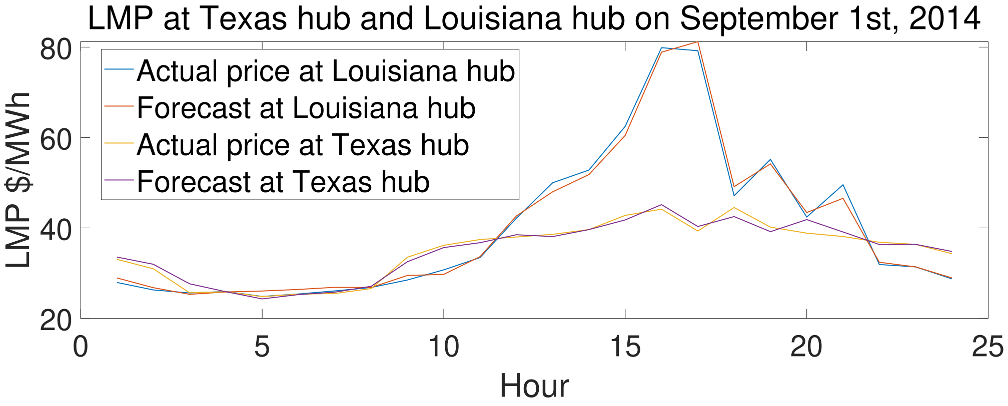


Figure 4.21: Ground-truth and Forecasted DALMPs for Hour-ahead Forecasting in Case 5.

Table 4.10: Average SMAPE (%) for Texas Hub in Case 5

Model	Naive (A)	NN	SVM	ARMA	GAN
SPA	0.53	0.67	0.59	0.69	0.78
SMAPE (%)	85	54	31	43	28

In Case 6, the effectiveness of the GAN-based model in day-ahead DALMP forecasting is evaluated by monthly MAPE ($MAPE_{month}$) [77]. January and July are selected (in [77]) to evaluate the forecasting results. The $MAPE_{month}$ of day-ahead DALMP forecasting obtained by different methods are listed in Table 4.11, which indicates the proposed GAN-based model outperforms all the other methods. Overall, The DALMP forecasting accuracy in January is low due to significant price spikes frequently seen in January. Compared to the second-best-performed method RS-SDA in [77], our proposed GAN model reduces the monthly MAPE in January by 6.02%. Moreover, the RS-SDA model is updated daily with the latest market data to maintain effectiveness [77]. In contrast, the proposed model does not require a daily model update, resulting in much less computational burden compared to RS-SDA.

Table 4.11: $MAPE_{month}$ (%) for Day-ahead Forecasting in Case 6

Model	Naive	NN	MARS	SVM	LASSO	SDA	RS-SDA	GAN
January	75.65	40.47	46.35	49.64	37.46	31.80	29.78	23.76
July	53.01	12.00	20.79	23.20	11.91	10.04	8.97	8.13

4.5 Conclusion

In this Chapter, a GAN-based approach is proposed to predict system-wide RTLMP. The proposed approach performs well across different types of price forecasting problems for various market participants. Historical public market data are organized into a general 3D tensor data structure, which stores the spatio-temporal correlations of the market data. The RTLMP prediction problem is then formulated as a 2D array prediction problem and solved using the proposed deep convolutional GAN model with multiple loss functions. The prediction accuracy is improved by an ARMA calibration approach to mitigate deviations caused by variation/uncertainty of generator bids. Case studies using real-world historical market data from ISO-NE, MISO, and SPP verify the performance of the proposed approach for both point-by-point price prediction accuracy and accuracy of capturing spatial correlations among prices at different locations.

Chapter 5

CONVOLUTIONAL LONG SHORT-TERM MEMORY-BASED GENERATIVE ADVERSARIAL NETWORK

In the wholesale electricity markets, LMPs are strongly spatio-temporal correlated. Most previous data-driven studies on LMP forecasting only leveraged temporal correlations among historical LMPs, and very few of them learned the spatial correlations to improve forecasting accuracy. This chapter proposes a convolutional long-short term memory (CLSTM)-based GAN to forecast LMPs from market participants' perspective. Historical LMPs of different price nodes are organized into a 3D tensor which stores the spatio-temporal correlations among LMPs. The LMP forecasting problem is formulated as a spatio-temporal sequence-to-sequence forecasting problem. The proposed approach is verified through case studies using public historical LMPs from MISO and ISO-NE, in comparison with other state-of-the-art LMP prediction approaches.

5.1 Introduction

Accurate price forecasting is crucial for market participants to make informed trading decisions in the wholesale electricity markets. Simultaneously, it is important to acknowledge that the strategic behaviors of market participants themselves directly influence electricity prices. The commonly accepted electricity pricing mechanism in the U.S. is the LMP, which is the dual variable derived in solving the DCOPF. During operations, LMPs are calculated by system operators based on the network topology, operation parameters, generation bids, and system demands. Given explicit knowledge of network topology and other system parameters, [12; 11] prove that

LMPs are a linear mapping of demands based on fixed generation cost functions. Therefore, LMPs at different price nodes are spatially correlated based on the grid topology and system-wide demand distributions. However, in real-world markets, generation cost functions are not fixed. Both generation bids and demands vary over time. Uncertainties in the time-varying generation bids (with bid-in values for generation cost functions) and demands jointly determine the temporal correlations among LMPs.

Nonetheless, it is worth noting that market participants do not have access to explicit information regarding physical network models or generation bids. In order to forecast locational marginal prices (LMPs) from the perspective of market participants, statistical approaches and data-driven methodologies that rely solely on publicly available market data have emerged as the primary candidate solutions.

Time-series-based statistical models are widely accepted to learn linear relationships in historical LMPs. The ARMAX model [37], ARIMA model [30], and AGARCH model [38] are applied to forecast LMPs for individual price nodes. These time-series-based statistical models completely ignore spatial correlations among historical LMPs.

Many data-driven LMP forecasting methods are developed based on learning SPR [82]. In [83; 48], various SPR-based approaches are applied to forecast LMPs using only publicly available historical market data. The LMP forecasting accuracy of these SPR-based approaches highly depends upon the nodal load forecasting accuracy and the assumption of fixed bidding strategies for all the generators. However, these conditions do not hold in real-world markets.

Machine learning methods, such as NNs [77; 84], LASSO [85], and long-short term memory (LSTM) [86; 87] are popular in LMP forecasting. In [86; 87], LSTM is shown to perform LMP forecasting effectively. However, this naive LSTM approach is only trained by learning temporal correlations. The forecasting accuracy can be further improved by considering advanced NN structures.

This chapter proposes a convolutional LSTM (CLSTM) based GAN to forecast system-wide LMPs from market participants' perspective, without power grid models and generator bidding details. The LMP forecasting problem is formulated as a sequence-to-sequence forecasting problem, and the LMP spatio-temporal correlations are learned by the CLSTM network during adversarial training. The proposed approach is trained to forecast the most likely future LMP sequences, which maximize the conditional probability given historical LMPs. Although the proposed CLSTM-based GAN model is applied to forecast system-wide LMPs, this model offers a general NN structure that can be utilized for other spatio-temporal forecasting problems in power systems, such as system-wide demand and wind/solar generation forecasting. In general, the LMP forecasting problem is more challenging than other power system forecasting problems (such as demand forecasting), since LMPs are highly volatile with price spikes. When conventional forecasting techniques are applied to LMP and demand forecasting problems, the typical forecasting errors could reach beyond 20% for LMP forecasting but around 2% for demand forecasting [88].

The rest of this chapter is organized as follows. Section 5.2 illustrates the sequence-to-sequence prediction formulation and the CLSTM-based predictor for LMP forecasting. Section 5.3 presents the GAN training procedure. Section 5.4 evaluates the proposed model through case studies with historical LMPs from MISO and ISO-NE. Section 5.5 concludes this chapter.

5.2 Sequence-to-Sequence LMP Forecasting

This section follows the general data structure we proposed in Chapter 3.2 to organize the historical LMPs into a 3D tensor. With this data structure, the CLSTM predictor in [89] is adopted to solve this spatio-temporal sequence LMP forecasting problem.

5.2.1 Data Structure and Normalization

Suppose historical hourly LMPs are collected over a geographical region represented by $N = m \times n$ price nodes for T consecutive hours. Following the general spatio-temporal data structure and normalization defined in Chapter 3.2, we preprocess and organize these historical data into a 3D tensor $\mathcal{X} \in \mathbb{R}^{m\times n\times T}$. The values of all elements in $\mathcal{X}$ are normalized to fall between -1 and 1. $\mathcal{X}$ consists of a sequence of 2D arrays, $\mathcal{X} = \{X^1, X^2, \cdots, X^T\}$. Each 2D array, X^t, contains system-wide LMPs from $N = m \times n$ price nodes at hour t.

5.2.2 Formulation of Sequence-to-Sequence Forecasting

The objective of LMP forecasting is to use historical LMP tensor to generate a forecasted tensor $\hat{\mathcal{Y}} \in \mathbb{R}^{m\times n\times K}$, which consists of a sequence of forecasted 2D arrays $\hat{\mathcal{Y}} = \{\hat{Y}^{T+1}, \hat{Y}^{T+2}, \cdots, \hat{Y}^{T+K}\}$. $\hat{\mathcal{Y}}$ contains the predicted LMPs at the same N price nodes for the following K hours. Let $\tilde{\mathcal{Y}} = \{X^{T+1}, X^{T+2}, \cdots, X^{T+K}\}$ denote the ground truth. A suitable predictor should generate/determine the most likely future LMP tensor given the historical LMP tensor $\mathcal{X}$, which can maximize the conditional probability below:

$$\hat{\mathcal{Y}} = \arg\max_{\mathcal{Y}} p(\mathcal{Y}|\mathcal{X}) \tag{5.1}$$

5.2.3 Convolutional Long-short Term Memory Network

To obtain the LMP predictor satisfying (5.1), the CLSTM network in [89] is adopted to build the structure of the LMP predictor. This CLSTM network is a multi-layer NN with many learnable parameters. During the training process using historical LMP data, these learnable parameters are optimally adjusted such that upon training convergence, the CLSTM network with the learned optimal parameters can generate the most likely future LMP tensors satisfying (5.1).

The CLSTM network is adopted for LMP forecasting since: 1) the CLSTM network is designed for general spatio-temporal sequence forecasting problems whose inputs and outputs are both spatio-temporal sequences/tensors, and the LMP forecasting problem (whose inputs and outputs are historical and future LMP tensors) falls within this general formulation; 2) the major advantage of the CLSTM network is that, compared to the classic LSTM network [55] which is widely acknowledged to be effective in time series forecasting (by learning temporal characteristics from historical datasets), the CLSTM network replaces the fully connected structures within the classic LSTM network by convolution operators (which are commonly used for learning spatial characteristics from historical datasets), enabling the CLSTM network for effectively capturing spatio-temporal correlations in historical LMPs. In [89], the CLSTM network solves a similar spatio-temporal precipitation sequence prediction problem over a geographic region.

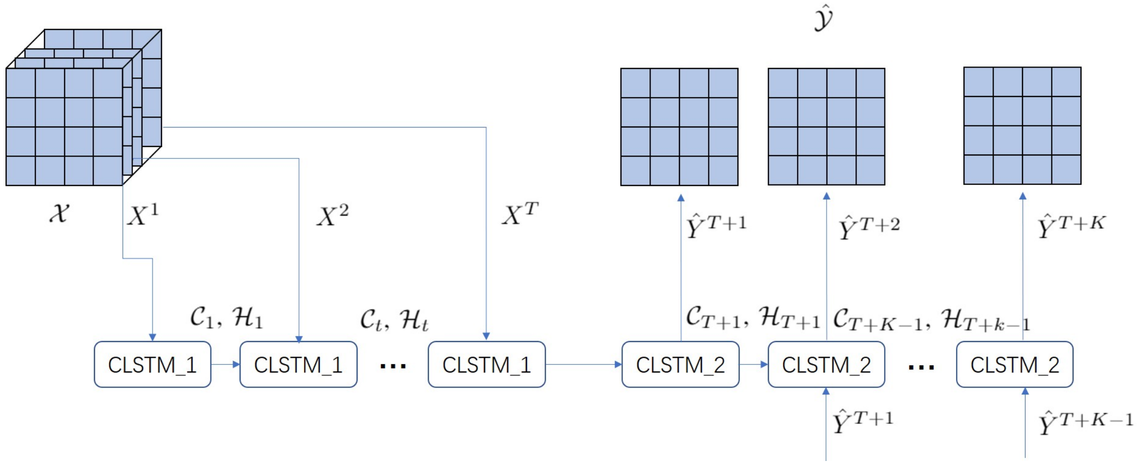


Figure 5.1: The Structure of the CLSTM-based LMP Predictor.

Our predictor architecture consists of two CLSTM networks, namely CLSTM_1 and CLSTM_2, with many learnable parameters, as shown in Fig. 5.1. The CLSTM_1 network runs T times to read in the input sequences. It extracts the spatio-temporal features of historical LMPs by transitioning the 3D input historical LMP tensor $\mathcal{X} = \{X^1, X^2, \cdots, X^T\}$ to the 3D hidden state (intermediate) tensor $\{\mathcal{H}_1, \mathcal{H}_2, \cdots, \mathcal{H}_T\}$ and the corresponding 3D output tensor $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_T\}$. This output tensor contains the extracted spatio-temporal features of historical LMPs. In the $t^{th}(t \geq 2)$ step, CLSTM_1 network takes two sets of inputs: 1) the 2D input historical LMP array X^t; 2) the 2D hidden state array $\mathcal{H}_{t-1}$ and the 2D output array $\mathcal{C}_{t-1}$ from the previous step. The CLSTM_2 network runs K times. It generates the forecasted LMP tensor $\hat{\mathcal{Y}} = \{\hat{Y}^{T+1}, \hat{Y}^{T+2}, \cdots, \hat{Y}^{T+K}\}$ following spatio-temporal features extracted by the CLSTM_1 network. In the $t^{th}(t \geq 2)$ step, CLSTM_2 network (for LMP prediction), takes two sets of inputs: 1) the 2D forecasted LMP array $\hat{Y}^{t-1}$; 2) the 2D hidden state array $\mathcal{H}_{t-1}$ and the 2D output array $\mathcal{C}_{t-1}$ from the previous step. In each running step, the CLSTM cell performs the following operations:

$$i_t = \sigma(W_{xi} * Z^t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \tag{5.2}$$

$$f_t = \sigma(W_{xf} * Z^t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \tag{5.3}$$

$$o_t = \sigma(W_{xo} * Z^t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_{t-1} + b_o) \tag{5.4}$$

$$\mathcal{C}_t = f_t \circ \mathcal{C}_{t-1} + i_t \circ tanh(W_{xc} * Z^t + W_{hc} * \mathcal{H}_{t-1} + b_c) \tag{5.5}$$

$$\mathcal{H}_t = o_t \circ tanh(\mathcal{C}_t) \tag{5.6}$$

where $Z^t = X^t$ and $Z^t = \hat{Y}^{t-1}$ for CLSTM_1 and CLSTM_2 networks, respectively; $*$, $\circ$, $\sigma(\cdot)$ are the convolution operator, Hadamard product, and sigmoid activation function, respectively; i_t, f_t, o_t are internal structures for each CLSTM network (the input, forget, and output gates, respectively); the weight matrices W_{xj} and biases b_j (with $j \in \{i, f, c, o\}$) are learnable parameters of the two CLSTM networks. In (5.5)-(5.6), for t^{th} step, the historical LMP array $Z^t = X^t$ in the CLSTM_1 network and the forecasted LMP array $Z^t = \hat{Y}^{t-1}$ (generated from $(t-1)^{th}$ step) in the CLSTM_2 network are transitioned to the hidden state array $\mathcal{H}_t$ and output array $\mathcal{C}_t$, through nonlinear operations in (5.2)-(5.6).

Within the aforementioned CLSTM networks, the consecutive 2D arrays are sequentially inputted and processed. Through this iterative process, the networks learn and capture the temporal correlations present in the data. Simultaneously, the spatial correlations within each iteration are captured through the inner convolution operations performed within the CLSTM networks. The internal structure details of the CLSTM cell are in [89].

As Fig. 5.1 shows, consecutive 2D arrays in the forecasted LMP tensor $\hat{\mathcal{Y}}$ are generated one by one, conditioned on the forecasted LMP arrays from the previous steps. Therefore, the conditional probability in (5.1) can be rewritten as the product of a series of conditional probabilities:

$$p(\mathcal{Y}|\mathcal{X}) = p(Y^{T+1}|\mathcal{X}) \prod_{i=T+2}^{T+K} p(Y^i|\mathcal{X}, Y^{T+1}, \cdots, Y^{i-1}) \tag{5.7}$$

During the training process, all learnable parameters of the two CLSTM networks, including the weight matrices W_{xj} and biases b_j (with $j \in \{i, f, c, o\}$) in (5.2)-(5.6), are optimized by minimizing the Euclidean distance between the forecasted and ground-truth 2D LMP arrays, $\hat{Y}^t$ and $\tilde{Y}^t$, using 2-norm loss:

$$\mathcal{L}_2(\hat{Y}, \tilde{Y}) = \ell_2(\hat{Y}, \tilde{Y}) = \left\|\hat{Y} - \tilde{Y}\right\|_2^2 \tag{5.8}$$

where $\|\cdot\|_2$ denotes the entry-wise 2-norm of a 2D array.

When training the CLSTM networks, simply minimizing the Euclidean distance (the 2-norm loss) over the dataset may lead to blurry prediction results which cannot maximize the conditional probability in (5.7), as discussed in [69]. Moreover, the 2-norm loss fails to consider the temporal correlations between consecutive forecasted 2D arrays $\hat{Y}^t$ and $\hat{Y}^{t+1}$, which could adversely affect the forecasting accuracy.

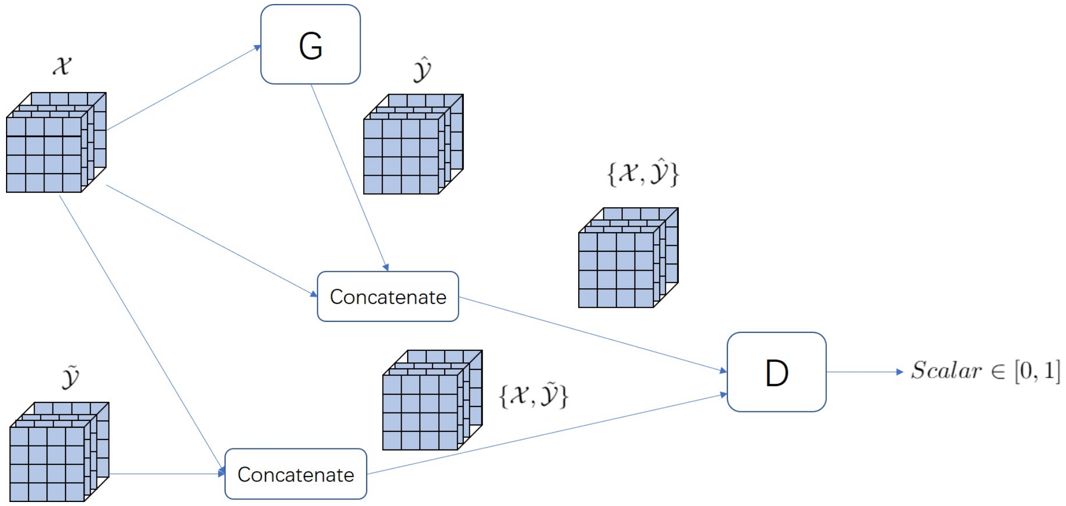


Figure 5.2: Architecture for Training the CLSTM-based GAN Model.

To solve these major drawbacks, the GAN model with multiple loss functions in Chapter 4.1 is adopted to train the predictor.

5.3 GAN Model for Sequence-to-Sequence Forecasting

In Chapter 4.1, an effective GAN model for price forecasting is proposed and evaluated. We take advantage of this GAN model and replace the original generator and discriminator with the CLSTM networks introduced above.

5.3.1 Sequence-to-Sequence Forecasting Model with GAN

The architecture for training the sequence-to-sequence predictor through the GAN model is shown in Fig. 5.2. In this architecture, the CLSTM-based predictor in Fig. 5.1 is employed as generator G; another CLSTM-based neural network is employed as discriminator D.

The generator G takes a sequence of 2D LMP arrays stored in tensor $\mathcal{X}$ as inputs, and generates a sequence of forecasted 2D LMP arrays as tensor $\hat{\mathcal{Y}}$. G is trained to maximize the conditional probability in (5.7), such that the forecasted tensor $\hat{\mathcal{Y}}$ is statistically similar to the ground-truth tensor $\tilde{\mathcal{Y}}$.

The discriminator D takes sequences $\{\mathcal{X}, \hat{\mathcal{Y}}\}$ or $\{\mathcal{X}, \tilde{\mathcal{Y}}\}$ as the input. D is trained to classify $\{\mathcal{X}, \hat{\mathcal{Y}}\}$ as fake and $\{\mathcal{X}, \tilde{\mathcal{Y}}\}$ as real. The output of D is a scalar between 0 and 1, indicating the probability of the input tensor being the ground truth.

The discriminator D and generator G are trained simultaneously; the weight matrices of D and G are updated iteratively, following the same training algorithm in Algorithm 1. The changes we make to D and G are described below, the rest of the details can be found in Chapter 4.1.

5.3.2 The Discriminator D

The discriminator D is a stacked CLSTM network followed by fully connected layers. The objective of D is to classify the input sequence $\{\mathcal{X}, \tilde{\mathcal{Y}}\}$ into class 1 (i.e., $\tilde{\mathcal{Y}}$ is classified as the ground-truth sequence) and the input sequence $\{\mathcal{X}, \hat{\mathcal{Y}}\} = \{\mathcal{X}, G(\mathcal{X})\}$ into class 0 (i.e., $\hat{\mathcal{Y}} = G(\mathcal{X})$ is classified as the generated fake sequence). The discriminator D is trained by minimizing the following distance function (loss function):

$$\mathcal{L}^{D}_{adv}(\mathcal{X}, \tilde{\mathcal{Y}}) = \mathcal{L}_{bce}(D(\{\mathcal{X}, \tilde{\mathcal{Y}}\}), 1) + \mathcal{L}_{bce}(D(\{\mathcal{X}, G(\mathcal{X})\}), 0) \tag{5.9}$$

where $\mathcal{L}_{bce}$ is the following binary cross-entropy:

$$\mathcal{L}_{bce}(k, s) = -[slog(k) + (1 - s)log(1 - k)] \tag{5.10}$$

where $k \in [0, 1]$ and $s \in \{0, 1\}$.

5.3.3 The Generator G

The generator G is a sequence-to-sequence predictor consisting of two CLSTMs as shown in Fig. 5.1. The input of G is a sequence of 2D LMP arrays for T hours; the output of G is a sequence of forecasted 2D LMP arrays for future K hours. During training process, the generator G is trained to minimize a certain distance between generated sequence $\hat{\mathcal{Y}}$ and the ground-truth sequence $\tilde{\mathcal{Y}}$ by the multi-loss function in (5.11), such that the conditional probability in (5.7) is maximized:

$$\mathcal{L}^G(\mathcal{X}, \tilde{\mathcal{Y}}) = \lambda_{adv}\mathcal{L}^G_{adv}(\mathcal{X}, \tilde{\mathcal{Y}}) + \lambda_{\ell_2}\mathcal{L}_2(\mathcal{X}, \tilde{\mathcal{Y}}) + \lambda_{gdl}\mathcal{L}_{gdl}(\mathcal{X}, \tilde{\mathcal{Y}}) + \lambda_{dcl}\mathcal{L}_{dcl}(\mathcal{X}, \tilde{\mathcal{Y}}) \quad (5.11)$$

where $\mathcal{L}^G(\mathcal{X}, \tilde{\mathcal{Y}})$ denotes the weighted multi-loss function for training G; $\mathcal{L}^G_{adv}(\mathcal{X}, \tilde{\mathcal{Y}})$, $\mathcal{L}_2(\mathcal{X}, \tilde{\mathcal{Y}})$, $\mathcal{L}_{gdl}(\mathcal{X}, \tilde{\mathcal{Y}})$ and $\mathcal{L}_{dcl}(\mathcal{X}, \tilde{\mathcal{Y}})$ denote four terms for this loss function (with the first and third terms explained separately in Chapter 4.1, the second term given in (5.8), and the last term explained in the following section); λ_{adv}, λ_{ℓ_2}, λ_{gdl}, λ_{gdl} denote hyperparameters for adjusting the weights of the four loss terms.

$\mathcal{L}_{dcl}(\mathcal{X}, \tilde{\mathcal{Y}})$ is the updated direction changing loss function added to $\mathcal{L}^G(\mathcal{X}, \tilde{\mathcal{Y}})$ to solve temporal correlation mismatches:

$$\mathcal{L}_{dcl}(\mathcal{X}, \tilde{\mathcal{Y}}) = \sum_{t=T+1}^{T+K} \sum_{i,j} |sgn(\hat{Y}^t_{i,j} - X^{t-1}_{i,j}) - sgn(\tilde{Y}^t_{i,j} - X^{t-1}_{i,j})| \quad (5.12)$$

where $sgn(\cdot)$ is the sign function:

$$sgn(z) = \begin{cases} -1 & if\ z \leq 0 \\ 0 & if\ z = 0 \\ 1 & if\ z \geq 0 \end{cases} \quad (5.13)$$

This direction changing loss function penalizes incorrect temporal correlations in the forecasted sequences during the training process, such that the correct temporal correlations are learned by G. This direction changing loss function along with other loss terms can effectively resolve the blurry prediction issues caused by using only the 2-norm loss function in (5.8).

5.4 Case Study

The proposed CLSTM-GAN model is tested using real-world LMP data from MISO [5] and ISO-NE [3]. The classic LSTM model in [86; 87] and other data-driven models, such as stacked denoising autoencoders (SDA) [77], standard neural network (NN), SVM, and Lasso are implemented as benchmark LMP forecasting models. The model performance is evaluated by calculating the mean absolute percentage error (MAPE) of the LMP forecasts. Because the proposed approach is a sequence-to-sequence forecasting method, it can be applied to forecast LMPs for different horizons. In this section, we will show 2 cases including both online forecasting and day-ahead forecasting.

Case 1

The training dataset contains hourly ISO-NE real-time LMP data of 9 price nodes in 2016 and 2017. The testing data is hourly ISO-NE real-time LMP data of the same price nodes in 2018. In Case 1(A), the proposed CLSTM-GAN model takes the tensor $\mathcal{X} \in \mathbb{R}^{3\times3\times6}$ for the past 6-hour LMPs to forecast the next hour's LMPs as $\hat{\mathcal{Y}} \in \mathbb{R}^{3\times3\times1}$. In Case 1(B), the proposed CLSTM-GAN model inputs the tensor $\mathcal{X} \in \mathbb{R}^{3\times3\times24}$ for the past 24-hour LMPs to forecast the next day's LMPs as $\hat{\mathcal{Y}} \in \mathbb{R}^{3\times3\times24}$.

Case 2

The training and testing dataset contains hourly MISO day-ahead LMP data of 9 price nodes from January 2012 to November 2014 (CLSTM-GAN is trained using data in 2012 and 2013). To compare with the benchmark models in [77], the testing dates used in [77] (1st, 10th, and 30th days in January, April, and August of 2014) are selected to demonstrate the forecasting accuracy. The proposed CLSTM-GAN model takes the tensor $\mathcal{X} \in \mathbb{R}^{3\times3\times24}$ for the past 24-hour LMPs as input to perform day-ahead forecasting.

5.4.1 Neural Network Architecture and Configurations

The proposed CLSTM-GAN model is implemented with Tensorflow [76] and trained on Google Colab using online GPU for acceleration. Table 5.1 lists the architecture details for G and D, where 'ConvLSTM2D' denotes the convolutional layer with CLSTM cells; 'Conv3DTranspose' denotes the convolutional transpose layer, and 'Dense' denotes the fully connected layer. In both cases, the generator G consists of 4 stacked 'ConvLSTM2D' layers (2 for CLSTM_1 and 2 for CLSTM_2); the discriminator D consists of 2 stacked 'ConvLSTM2D' layers and 3 'Dense' layers.

All the 'ConvLSTM2D' and 'Conv3DTranspose' layers in G are followed by batch normalization layers and ReLU units, while the 'ConvLSTM2D' and 'Dense' layers in D are followed by batch normalization layers, Leaky-ReLU units, and dropout layers. In all the cells of the 'ConvLSTM2D' layers, the recurrent activation functions are set as *sigmoid* function.

In G, the kernel size of all CLSTM cells is 3×3, the stride size of all CLSTM cells is 1×1; the kernel size of all 'Conv3DTranspose' layers is $1 \times 3 \times 3$, the stride size of all 'Conv3DTranspose' layers is $1 \times 1 \times 1$; all the 'ConvLSTM2D' and 'Conv3DTranspose' layers are padded. In D, the kernel size and stride size are the same as those used in G, but all the 'ConvLSTM2D' layers are not padded; the dropout rates are set to 0.3, the small gradients are set to 0.2 when the Leaky-ReLU is not active. In all models, a standard SGD optimizer is utilized for adversarial training. In the training process, the minibatch size is set to 4 for all models. In Case 1(A), the learning rates for G and D are 0.005 and 0.001, respectively. In Case 1(B) and Case 2, learning rates for G and D are 0.0005 and 0.0001, respectively. For all CLSTM-GAN models, the

Table 5.1: Neural Network Architecture Details

Case 1 (A)	**Generator G** (Layer Type, Feature Map)	**Discriminator D** (Layer Type, Feature Map)
Input	$3 \times 3 \times 6$	$3 \times 3 \times 7$
Layer 1	ConvLSTM2D, 64	ConvLSTM2D, 64
Layer 2	ConvLSTM2D, 256	ConvLSTM2D, 256
Layer 3	ConvLSTM2D, 128	Dense, 1024
Layer 4	ConvLSTM2D, 64	Dense, 512
Layer 5	Conv3DTranspose, 1	Dense, 256
Output	$3 \times 3 \times 1$	scalar$\in [0, 1]$

Case 1 (B) **Case 2**	**Generator G** (Layer Type, Feature Map)	**Discriminator D** (Layer Type, Feature Map)
Input	$3 \times 3 \times 24$	$3 \times 3 \times 48$
Layer 1	ConvLSTM2D, 64	ConvLSTM2D, 64
Layer 2	ConvLSTM2D, 128	ConvLSTM2D, 256
Layer 3	ConvLSTM2D, 256	Dense, 1024
Layer 4	ConvLSTM2D, 64	Dense, 512
Layer 5	Conv3DTranspose, 1	Dense, 256
Output	$3 \times 3 \times 24$	scalar$\in [0, 1]$

hyperparameters in (5.11) are set to $\lambda_{adv} = \lambda_{dcl} = 0.2$, $\lambda_{\ell_2} = \lambda_{gdl} = 1$. More details on the model structures, the adversarial training algorithm, and the parameters not listed in this section are given in Chapter 4.1.

5.4.2 Case Study Results

In Case 1(A) and Case 1(B), the trained CLSTM-based generator G, the naive CLSTM model, and the classic LSTM model are employed to perform online forecasting (Case 1(A)) and day-ahead forecasting (Case 1(B)), respectively, for ISO-NE's hourly LMPs in 2018. Fig. 5.3 shows the LMPs forecasted by G and the ground-truth LMPs at ISO-NE's VT price node in 2018 for Case 1(A). Fig. 5.3 demonstrates the forecasted LMPs successfully capture the temporal characteristics among LMPs of ISO-NE in 2018. The MAPEs of both online forecasting and day-ahead forecasting are computed and compared in Table 5.2. Table 5.2 demonstrates that the proposed CLSTM-GAN model generally has better forecasting performance compared to the classic LSTM model and the naive CLSTM model for both online forecasting and day-ahead forecasting. In day-ahead forecasting, the LMPs in the forecasted tensor are jointly conditioned on the historical input LMP tensor and the previously forecasted LMP arrays. This will cause the forecasting errors to be accumulated during the day-ahead forecasting process, leading to larger MAPEs in the day-ahead forecasting

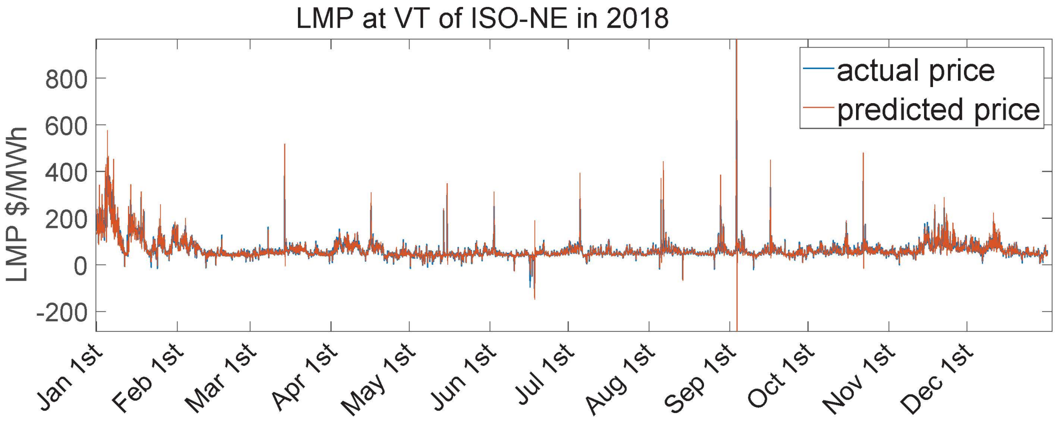


Figure 5.3: Ground-truth and Forecasted LMPs at the VT Price Node in ISO-NE for Case 1(A).

results compared to MAPEs in the online forecasting results, as shown in Table 5.2.

Table 5.2: LMP Forecasting Errors in Case 1

Case	**GLSTM-GAN**	**GLSTM**	**LSTM**
MAPE (%) in Case 1(A)	10.23	12.35	13.82
MAPE (%) in Case 1(B)	15.07	23.79	27.41

To further evaluate the proposed CLSTM-GAN model, the state-of-the-art SDA and RS-SDA [77] models and other popular data-driven approaches are included as benchmark models in Case 2. All models are evaluated with LMPs from the Indiana Hub in MISO. Table 5.3 shows day-ahead forecasting MAPEs obtained using different approaches on the same testing dataset.

Table 5.3: LMP Forecasting Errors in Case 2

Case	**LSTM**	**NN**	**MARS**	**SVM**	**Lasso**	**SDA**	**RS-SDA**	**CLSTM-GAN**
MAPE(%)	18.03	18.65	25.59	20.47	21.63	12.87	12.18	11.53

In Table 5.3, the proposed CLSTM-GAN model (with a MAPE of 11.53%) outperforms the state-of-the-art RS-SDA forecasting model (with a MAPE of 12.18%) and all the other benchmark models. The RS-SDA model in [77] has the second-best performance. However, this RS-SDA model needs to be updated/re-trained daily, causing a significant computational/training burden. Besides, each well-trained RS-SDA model can only forecast LMPs for one price node. Therefore, forecasting system-wide LMPs requires a large number of RS-SDA models. Compared to the RS-SDA model, our CLSTM-GAN model can forecast system-wide LMPs for a long time period without daily updates, which is much more efficient.

5.5 Conclusion

This chapter proposes a CLSTM-GAN approach to forecast system-wide LMPs from market participants' perspective in online and day-ahead manners. Spatio-temporal correlations among historical LMPs are learned by the CLSTM network during adversarial training. In case studies using ISO-NE's and MISO's public data, the proposed method outperforms the state-of-the-art benchmarks and other data-driven approaches.

Chapter 6

DECISION TRANSFORMER BASED SPATIOTEMPORAL FORECASTING

System-wide LMPs are jointly determined by the interactions between the market clearing process and the generator bidding process. These two processes involve sequential decision-making, with the objective of achieving optimal market clearing decisions and strategic bidding decisions. The market clearing process determines the allocation of electricity supply and demand at specific locational nodes, while the generator bidding process involves the strategic submission of bids by generators to maximize their profits. The outcome of these sequential decision-making processes directly influences the system-wide LMPs. In this chapter, the problem of forecasting market participants' RTLMPs is formulated by considering the interactive market clearing and generation bidding decision-making processes as a sequential decision-making model. A special NN model, named the spatio-temporal decision transformer (ST-DT), is proposed to learn the underlying sequential decision-making model from historical spatio-temporal market data (including locational demands, LMPs, and fuel prices) and forecast RTLMPs as the future actions of these interactive decision-making processes. This spatio-temporal decision transformer is constructed using stacked spatio-temporal multi-head self-attention mechanisms (ST-MHSA). This architecture provides flexibility in addressing a wide range of sequential decision-making problems by enabling selective attention to different data at various locations and timestamps. This algorithm can also be applied to other data-driven sequential decision-making problems in power systems, which are originally solved by offline reinforcement learning (RL) with high computation complexity in training. Case studies on the data sets from SPP and ISO-NE, verify that the proposed approach can forecast LMPs and learn the spatio-temporal correlations accurately.

6.1 Introduction

In the U.S. electricity markets, LMPs are settled twice in the DA and RT markets [9; 90]. In each market, system operators will collect locational generation bids from generators and locational demands from market participants as inputs to run OPF, and calculate LMPs. As a forward market, DA market is cleared based on DA generation bids and DA demand forecast before the operating day, and the DALMP will be published one day before the operating day. As a spot market, the RT market will clear the difference between DA commitments and the actual RT demand based on separate, second RTLMPs, which are either paid or charged to participants in the DA market for demand or generation that deviates from the DA commitments [9; 10]. To maximize their profits, generators will decide their RT bids based on both DA market data and RT market data. This generator RT bidding process can be considered a decision making procedure. Considering the DA market data as the previous states and RT demand data as the current states, the generator RT bid is the decision made based on previous states and current states. Once the generator bids are determined and submitted to the RT market, RTLMPs can be computed as the dual variables derived from solving the deterministic OPF in RT market [11; 12], which is the decision making process for optimal RT market clearing. Therefore, the RTLMPs can be viewed as the consequence of the interactive decision making process of generator bidding and RT market clearing (with deterministic OPF).

One drawback of the above machine learning (ML)-based forecasting approaches [55; 56; 57; 91; 92; 93] is the slow training speed and difficulty of convergence. For example, LSTM-based approaches have to read in the input sequence step by step, process, and pass the hidden state one by one. GAN-based approaches indeed come with their challenges during the training process and often require careful tuning to achieve desirable results. The training of GANs involves finding a balance between the generator and discriminator networks, ensuring that both networks learn effectively without one overpowering the other. Issues such as mode collapse, instability, and vanishing gradients can arise and hinder the training process. Proper network architecture design, selection of loss functions, learning rate scheduling, and regularization techniques are crucial for successful training of GANs. Additionally, hyperparameter tuning and extensive experimentation might be necessary to optimize the performance of GAN-based approaches. Recently, the attention mechanism-based approach [94] is proved powerful in processing sequence data with better interpretability. Compared with conventional sequence models, such as RNN and LSTM, the attention mechanism can process all the time steps of the input sequence simultaneously. The parallel operation can speed up the computation significantly. In [95], an attention temporal convolutional network is proposed by combining CNN with an attention mechanism to extract spatio-temporal renewable generation features from neighboring sites. This architecture is implemented for the ultra-short-term spatio-temporal forecasting of renewable resources. Besides forecasting renewable generation, the self-attention (SA) mechanism is combined with LSTM to forecast LMPs on a day-ahead basis in [96; 97]. Working as encoders, LSTM models in [96; 97] generate a representation of the input sequence as the input of the following attention mechanism. These attention LSTM approaches only account for temporal correlation. They ignore spatial dependencies of LMPs caused by the uncertainties of locational supply and locational demand.

For market participants who trade electricity over multiple price nodes (locations), an ideal RTLMP predictor should accurately predict prices at different nodes over a wide area. The spatio-temporal correlations among RTLMPs are expected to be captured by a data-driven approach. Previous approaches usually ignore spatial dependencies among locational demands and locational supplies or make improper assumptions by ignoring generator bid variations. The example in [98] demonstrates that even a slight variation of either the load distribution or bidding strategy will cause significant changes in LMP components. We believe that the variation of generator bids should be taken into consideration to account for the spatio-temporal dependencies in prediction. However, in most electricity markets, generation bidding data are confidential or published with several months' delay [9; 90]. To properly incorporate the implicit bid variation, the natural gas price is considered as a rough representation of generator bids in the overall decision making process of generator bidding and market clearing. Organizing all the public market data as a market data sequence, this chapter proposes a generic forecasting approach to learn the spatio-temporal dependencies among market data sequences and forecast RTLMPs accurately.

The rest of the chapter is organized as follows. Section II defines the multi-channel market data sequence, and formulates the RTLMP forecasting as a sequence modeling problem; Section III proposes the ST-DT-based RTLMP prediction model, consisting of stacked multiple multi-head attention mechanisms; Section IV presents case studies and compares the performance of the proposed approach with several benchmark methods; Section V concludes this chapter.

6.2 Problem Formulation

Consider a set of publicly available historical hourly market data published by the system operator for N nodes and T consecutive hours. These market data include demand data, LMP data, and generation bid data for both the RT market and the DA market.

Definition 4 *Let* $D_{DA}^{t,n}$, $D_{RT}^{t,n}$, $LMP_{DA}^{t,n}$, $LMP_{RT}^{t,n}$, $B_{DA}^{t,n}$, $B_{RT}^{t,n}$, *represent the historical DA demand data point, RT demand data point, DALMP data point, RTLMP data* *point, DA bid data point and RT bid data point collected from node* $n \in \{1, 2, \cdots, N\}$ *at time* $t \in \{1, 2, \cdots, T\}$*, respectively. These spatio-temporal historical market data points are also called* ***tokens*** *in this paper.*

Definition 5 *At each time* t*, the market is described by several* ***states***, $D^t_{DA} = \{D^{t,1}_{DA}, D^{t,2}_{DA}, \cdots D^{t,N}_{DA}\} \in \mathbb{R}^N$, $D^t_{RT} = \{D^{t,1}_{RT}, D^{t,2}_{RT}, \cdots D^{t,N}_{RT}\} \in \mathbb{R}^N$, $LMP^t_{DA} = \{LMP^{t,1}_{DA}, LMP^{t,2}_{DA}, \cdots LMP^{t,N}_{DA}\} \in \mathbb{R}^N$*, and* $B^t_{DA} = \{B^{t,1}_{DA}, B^{t,2}_{DA}, \cdots B^{t,N}_{DA}\} \in \mathbb{R}^N$*. These states contain DA demands, RT demands, DALMPs, and DA bids across multiple nodes, respectively. In the RT generator bidding decision making process, the* ***actions*** *of generators across multiple nodes at each time* t *are the RT bid decisions* $B^t_{RT} = \{B^{t,1}_{RT}, B^{t,2}_{RT}, \cdots B^{t,N}_{RT}\} \in \mathbb{R}^N$.

Definition 6 *The RT OPF process can be represented by a function whose output* LMP^T_{RT} *is the RTLMP:*

$$LMP^T_{RT} = OPF(D^T_{RT}, B^T_{RT}) \tag{6.1}$$

This equation indicates the OPF process defines an implicit mapping $(D^T_{RT},\ B^T_{RT}) \rightarrow LMP^T_{RT} = \{LMP^{T,1}_{RT},\ LMP^{T,2}_{RT},\ \cdots\ LMP^{T,N}_{RT}\} \in \mathbb{R}^N$*, which represents the market clearing decision making process at time* T*, based on the inputs of RT demands and bids. The output* LMP^T_{RT} *contains system-wide RTLMPs at time* T*, which can be considered as the* ***action*** *out of this decision-making process. Such* ***action*** *should be forecasted at time* T *from the market participants' perspective.*

Consider the generator bidding strategies in RT market as a sequential decision making process described by the tuple $(\mathcal{D}_{DA}, \mathcal{B}_{DA}, \mathcal{LMP}_{DA}, \mathcal{D}_{RT}, \mathcal{B}_{RT})$, where $\mathcal{D}_{DA} = \{D^1_{DA}, D^2_{DA}, \cdots D^T_{DA}\}$, $\mathcal{B}_{DA} = \{B^1_{DA}, B^2_{DA}, \cdots B^T_{DA}\}$, $\mathcal{LMP}_{DA} = \{LMP^1_{DA}, LMP^2_{DA}, \cdots LMP^T_{DA}\}$, $\mathcal{D}_{DA} = \{D^1_{RT}, D^2_{RT}, \cdots D^T_{RT}\}$, $\mathcal{B}_{RT} = \{B^1_{RT}, B^2_{RT}, \cdots B^T_{RT}\}$. The decision process consists of previous states $D^t_{DA} \in \mathcal{D}_{DA}$ (DA demands), $LMP^t_{DA} \in$

$\mathcal{LMP}_{DA}$ (DALMPs), $B^t_{DA} \in \mathcal{B}_{DA}$ (DA bids), current states $D^T_{RT} \in \mathcal{D}_{RT}$ (RT demands), and actions $B^T_{RT} \in \mathcal{B}_{RT}$ (RT bids), where $t \in \{1, 2, \cdots, T\}$.

In this sequential decision-making process, generators across the market decide the RT bids based on states and actions of previous T hours. Because the latest bidding information B^T_{RT} is not available, market participants have to estimate generators' implicit decision-making process for RT bids by maximizing the conditional probability:

$$\hat{B}^T_{RT} = \arg\max_{\tilde{B}^T_{RT}} p(\tilde{B}^T_{RT} = B^T_{RT} | D^{1\sim T}_{DA}, B^{1\sim T}_{DA}, D^{1\sim T}_{RT}, B^{1\sim T-1}_{RT}) \tag{6.2}$$

From market participants' perspective, the latest system-wide bidding information in the transition dynamics in (6.2) is not publicly available. Most system operators publish this information with months of delays [9; 90]. During market participants' RT bidding decision-making process, this information can be replaced by system-wide public LMP information, which is an explicit representation of bidding information as shown by the inverse process of (6.1). Substituting historical bidding data $B^{1\sim T}_{DA}$ and $B^{1\sim T-1}_{RT}$ with historical LMP data $LMP^{1\sim T}_{DA}$ and $LMP^{1\sim T-1}_{RT}$, Equation (6.2) becomes:

$$\hat{B}^T_{RT} = \arg\max_{\tilde{B}^T_{RT}} p(\tilde{B}^T_{RT} = B^T_{RT} | D^{1\sim T}_{DA}, LMP^{1\sim T}_{DA}, D^{1\sim T}_{RT}, LMP^{1\sim T-1}_{RT}) \tag{6.3}$$

Equation (6.3) represents the market participants' RT bidding decision-making process which determines the proper RT bids at time T based on system-wide historical RT and DA demands and LMPs. Substituting the ground-truth B^T_{RT} in (6.1) with the estimated $\hat{B}^T_{RT}$ in (6.3), the RTLMP forecasting problem can be formulated as a combination of both the OPF-based market clearing decision marking process and the market participants' RT bidding decision-making process. Since the RT bidding decisions $\hat{B}^T_{RT}$ are actions of the sequential decision-making process in (6.3), the RTLMPs at time T in (6.1) can then be viewed as the outputs/actions of two interactive sequential decision-making process in (6.1) and (6.3), given the historical states of RT demands, DA demands, RTLMPs, and DALMPs in (6.1) and (6.3). Therefore, the market participants' learning goal for RTLMP forecasting is to learn the meaningful pattern of the following trajectory representation:

$$\begin{aligned}\tau =&(D_{DA}^{1}, LMP_{DA}^{1}, D_{RT}^{1}, LMP_{RT}^{1}, D_{DA}^{2}, LMP_{DA}^{2},\\ &D_{RT}^{2}, LMP_{RT}^{2}, \cdots, D_{DA}^{T}, LMP_{DA}^{T}, D_{RT}^{T}, LMP_{RT}^{T})\end{aligned} \tag{6.4}$$

where τ is an ordered market data sequence, which can be considered a sequential decision-making process for the interactive RT bidding and market clearing problems. LMP_{RT}^{T} is the action to be forecasted based on the states prior to its time step. In the sequential decision making process, τ, historical states $D_{DA}^{t,n}$, $D_{RT}^{t,n}$, $LMP_{DA}^{t,n}$, $LMP_{RT}^{t,n}$ at different time steps t and different nodes n contribute differently to the action LMP_{RT}^{T}. The state's position in the sequence τ should be utilized as conditional information for learning. The generator's bidding strategies B_{RT}^{t} and B_{DA}^{t} are also highly related to fuel prices [10]. Therefore, both the position information and fuel price information should be properly encoded and embedded into sequence τ. Fig. 6.1 shows the overall decision-making process of the wholesale electricity market. This overall decision-making process consists of bidding decision process (BDP in Fig. 6.1) and market clearing process (OPF Fig. 6.1). A well-trained LMP predictor should learn the underlying model for this sequential decision-making process from the historical state-action trajectory, such that the learned decision-making model can accurately predict the next action (system-wide LMPs) in the sequential decision-making process, which minimizes the distance between the predicted LMPs and actual LMPs. This sequential decision-making forecasts the next action by maximizing conditional probability $p(\hat{LMP}_{RT}^{T} = LMP_{RT}^{T}|\tau')$, where historical market data sequence τ' is the ordered sequence τ excluding LMP_{RT}^{T}.

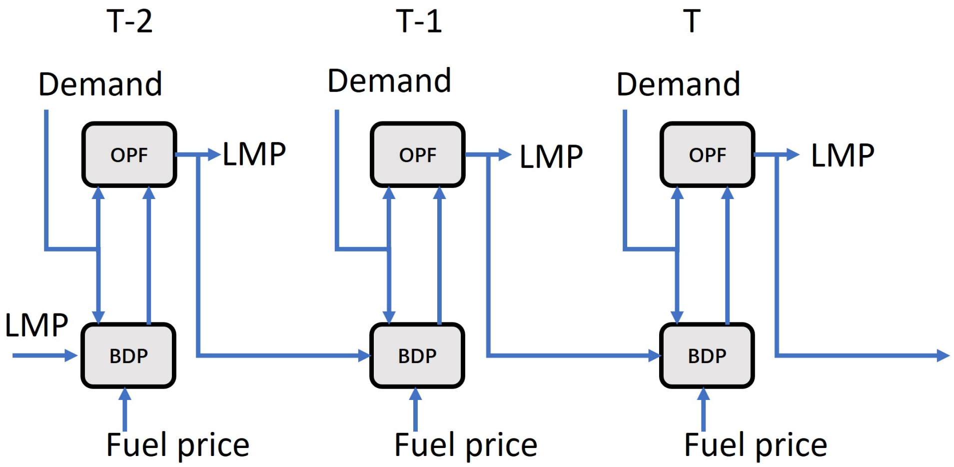


Figure 6.1: Overall Decision-making Process of the Wholesale Electricity Market.

6.3 Spatio-Temporal Decision Transformer

Built upon the above sequential decision-making formulation using system-wide state-action trajectory with spatio-temporal market data, a DT-based predictor is adopted to learn the underlying system-wide decision marking model by 1) extracting the spatio-temporal dependencies of the state-action trajectory; and 2) assigning different weights to the historical market data tokens, which are spatio-temporal correlated, to determine the mapping $\mathbb{R}^{(4T-1)\times N} \rightarrow \mathbb{R}^{1\times N}$ to predict next action (system-wide LMPs) across the market.

The transformer NN architecture is initially proposed to efficiently model sequence data [94]. The transformer NN is further modified to decision transformer NN to perform data-driven sequential decision making. In [99], the decision transformer NN is proposed to solve the data-driven sequential decision-making problem (formulated as an offline RL problem) via conditional sequence modeling. Without the need for dynamic programming (the foundation of RL), decision transformer NN can model data sequence to perform policy optimization of decision-making problems on offline

RL benchmarks. This allows the decision transformer to take advantage of parallel processing to achieve less training time and better convergence. However, the original decision transformer is only designed to process sequential information, ignoring spatial information processing. This limits its applications toward solving system-wide sequential decision-making problems (such as system-wide LMP forecasting) which require both sequential and spatial information processing. In [95], CNN is combined with the attention mechanism to extract spatial features before learning the temporal correlations. This attention convolutional network will affect the parallel processing and the convergence of the attention network. In this section, we propose an ST-DT network as shown in Fig. 6.2, consisting of input encoder, temporal multi-head self-attention (T-MHSA) network, spatial multi-head self-attention (S-MHSA) network, and output decoder, to learn spatio-temporal correlations among input tokens and solve the system-wide RTLMP forecasting problem which is formulated as a data-driven sequential decision making problem in the previous section. This allows us to leverage the simplicity and scalability of the decision transformer NN architecture to reduce training time and complexity without degrading the RTLMP forecasting accuracy.

6.3.1 Input Encoder Neural Network

The inputs of the proposed ST-DT are first processed and mapped to an embedded sequence τ by an input encoder, as shown in Fig. 6.2, including input embedding and timestamp encoding. In Fig. 6.2, the red box represents adding embedded fuel price tokens to the embedded input sequence; the blue box represents adding embedded timestamp information to the embedded input sequence. Then, the sequence τ is divided into N locational sequences τ_n, $\forall n \in \{1, \cdots, N\}$.

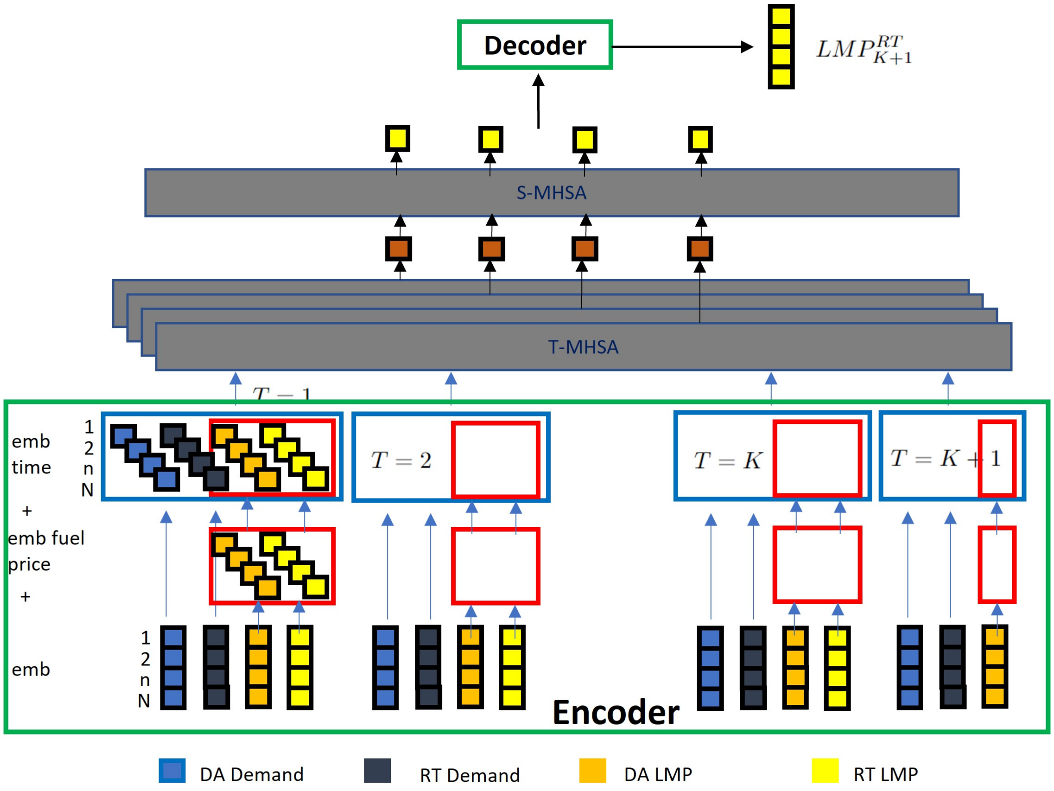


Figure 6.2: Illustration of the Processing of Spatio-temporal Correlated Input Sequences in the Spatio-temporal Decision Transformer.

Input Embedding

The decision transformer takes 5 sequences as input: DA demand sequence $D_{DA}^{1\sim T} \in \mathbb{R}^{T\times N}$, DALMP sequence $LMP_{DA}^{1\sim T} \in \mathbb{R}^{T\times N}$, RT demand sequence $D_{RT}^{1\sim T} \in \mathbb{R}^{T\times N}$, RTLMP sequence $LMP_{RT}^{1\sim T-1} \in \mathbb{R}^{(T-1)\times N}$, fuel price sequence $FP \in \mathbb{R}^{T\times 1}$. The corresponding tokens $LMP_{DA}^{t,n}$, $LMP_{RT}^{t,n}$, $D_{DA}^{t,n}$, $D_{RT}^{t,n}$, FP^t in the input sequences are not within the same ranges and dimensions. Each input token is projected to the same dimension M and range $[-1,1]$ via the input embedding model: $LMP_{(\cdot)e}^{t,n} = Embedding_{LMP}(LMP_{(\cdot)}^{t,n}) \in \mathbb{R}^M$, $D_{(\cdot)e}^{t,n} = Embedding_D(D_{(\cdot)}^{t,n}) \in \mathbb{R}^M$, $FP_e^t = Embedding_{FP}(FP^t) \in \mathbb{R}^M$, where $(\cdot) = DA$ or RT, $Embedding_{LMP}()$, $Embedding_D()$, $Embedding_{FP}()$ are the learnable projection models built upon fully connected (FC)

NNs.

The embedded input sequence of market data tokens ($LMP^t_{(\cdot)e}$, $D^t_{(\cdot)e}$) are organized into an embedded sequence following the order of trajectory representation in (6.4). At each time step t, the embedded fuel price token FP^t_e is added to the corresponding LMP tokens in the embedded sequence: $LMP^{t,n}_{(\cdot)ep} = LMP^{t,n}_{(\cdot)e} + FP^t_e$.

Timestamp Encoding

Because the proposed model processes the sequence parallelly, there is no recurrence or convolution operation to account for sequential information in the input sequence. To make use of the sequential order of the sequence, time step information is added to the embedded input sequence via timestamp encoding. The encoded time step information $TE(t) \in \mathbb{R}^M$ has the same dimension M, therefore it can be added to the corresponding embedded input tokens, where $TE()$ is the learned timestamp embedding function in [99]. In this paper, we use sine and cosine functions for $TE()$, which are also known as positional encoding functions. More details on these encoding functions can be found in [94].

The output is a data sequence containing sequential market information of all nodes:

$$\begin{aligned}\tau_e =& (D^1_{DAet}, LMP^1_{DAet}, D^1_{RTet}, LMP^1_{RTet}, D^2_{DAet}, LMP^2_{DAet}, D^2_{RTet}, LMP^2_{RTet}, \\ & \cdots, D^T_{DAet}, LMP^T_{DAet}, D^T_{RTet})\end{aligned} \tag{6.5}$$

where $\tau_e \in \mathbb{R}^{(4T-1)\times N\times M}$; $D^t_{(\cdot)et} = \{D^{t,1}_{(\cdot)e}, \cdots, D^{t,N}_{(\cdot)e}\} + TE(t)$, $LMP^t_{(\cdot)et} = \{LMP^{t,1}_{(\cdot)ep}, \cdots, LMP^{t,N}_{(\cdot)ep}\} + TE(t)$, $(\cdot) = DA$ or RT.

Locational Sequence

The overall encoded market data sequence τ_e is reorganized into N locational sequences as the ultimate output of the input encoder:

$$\begin{aligned}\tau_n =&(D_{DAet}^{1,n}, LMP_{DAet}^{1,n}, D_{RTet}^{1,n}, LMP_{RTet}^{1,n}, D_{DAet}^{2,n}, LMP_{DAet}^{2,n}, D_{RTet}^{2,n}, LMP_{RTet}^{2,n},\\ &\cdots, D_{DAet}^{T,n}, LMP_{DAet}^{T,n}, D_{RTet}^{T,n})\end{aligned} \tag{6.6}$$

Each of these locational sequences will be taken as the input of a T-MHSA, as shown in Fig. 6.3, to extract temporal features for the corresponding node.

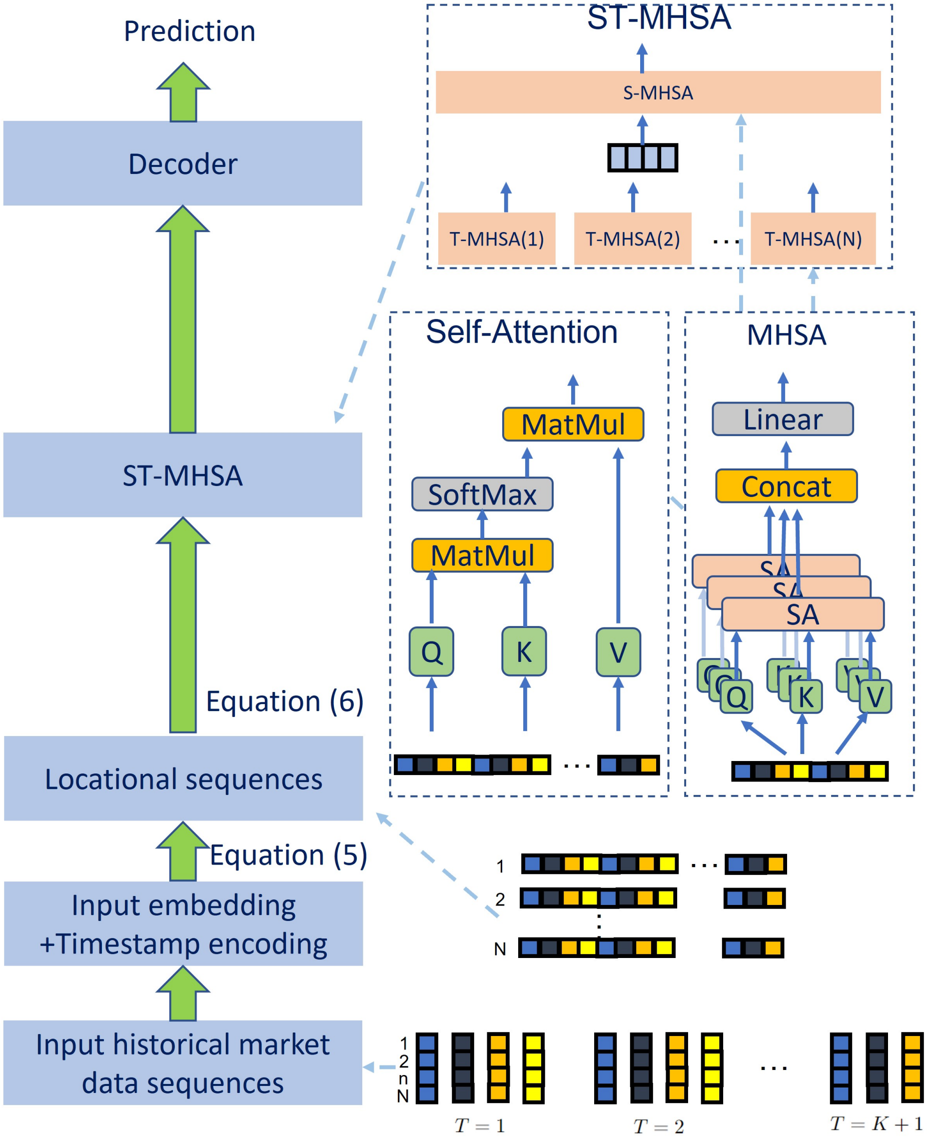


Figure 6.3: Overview of the Proposed Spatio-temporal Forecasting Method. The Left Column Is the Overall Architecture of the Stacked Decision Transformer-based Approach. The Right Column Demonstrates the Inner Details of the Spatio-temporal Multi-head Self-attention Mechanism. The ST-MHSA Consists of a Series of Parallel T-MHSAs Followed by an S-MHSA. The Temporal Correlation among Historical States Is Learned by T-MHSAs, and the Spatial Correlation among Learned Hidden States from T-MHSAs Is Captured by S-MHSAs.

6.3.2 Spatio-Temporal Multi-Head Self-Attention Neural Network

The traditional decision transformer is built upon multi-head self-attention (MHSA) mechanism, which cannot process spatial information effectively. All spatial features at the same time step are assigned the same weight by the traditional MHSA. The Spatial correlations hidden in the input data are ignored by MHSA. We construct an ST-MHSA network, consisting of a series of T-MHSAs followed by an S-MHSA, to process both temporal and spatial information of the input sequences, as shown in Fig. 6.3. Both T-MHSA and S-MHSA are designed to learn the correlations among input sequential states. In our design, as shown in Fig. 6.3, the correlation among T-MHSA's input sequential states contains temporal information; the correlation among S-MHSA's input sequential states contains spatial information. Overall, the proposed ST-MHSA network is more effective and flexible in learning the spatio-temporal correlated data.

Multi-Head Self-Attention Neural Network

The MHSA network is the core architecture of T-MHSA and S-MHSA. MHSA network reads in the embedded input sequence and generates an output sequence. The MHSA's output is a combination of data points in its input sequence. The weights of the learned combination are determined by the importance of each token in the sequence to the last token. These learnable weights represent the different strengths of the sequential correlations (temporal correlations for T-MHSA and spatial correlations for S-MHSA) between various historical market data and the forecasted prices. Because the attention mechanism is powerful in processing sequence data simultaneously, the MHSA network is applied to conduct parallel operation to speed up the computation of weights and reduce the training time significantly. Let $X \in \mathbb{R}^{L\times F}$ be the input sequence of an MHSA network, where L is the length of this sequence, F is the feature dimension. The input sequence X is projected to three new representations via three independent learnable linear transformation matrices [94]:

$$Q = X \cdot W_Q \tag{6.7}$$

$$K = X \cdot W_K \tag{6.8}$$

$$V = X \cdot W_V \tag{6.9}$$

where $Q, K, V \in \mathbb{R}^{L \times d}$ are also known as **Query**, **Key** and **Value** [94], respectively; $W_Q, W_K, W_V \in \mathbb{R}^{F \times d}$ are the learnable transformation matrices. The SA mechanism generates a new sequence $\hat{Y}$, in which each token is the weighted combination of all tokens of the input sequence X [94]:

$$\hat{Y} = SA(Q, K, V) = softmax(Q \cdot K^T)V \tag{6.10}$$

where $\hat{Y} \in \mathbb{R}^{L \times d}$; $softmax(Q \cdot K^T)$ is the attention matrix, whose elements are attention scores. These attention scores are the learned weights of the combination on the values in V, and represent the corresponding importance of input tokens to the output.

To enhance the performance of SA, multiple SAs are stacked to an MHSA by concatenating the output of each SA to calculate an aggregated result:

$$MHSA(Q, K, V) = Concat(head^1, \cdots, head^h)W^O \tag{6.11}$$

where,

$$head^i = SA(Q^i, K^i, V^i), \forall i \in \{1, \cdots, h\} \tag{6.12}$$

where, Q^i, K^i and V^i are projected by (6.7)-(6.9) via corresponding transformation matrices W_Q^i, W_K^i and W_V^i, respectively; h is the number of attention heads; $W^O \in \mathbb{R}^{(h \cdot d) \times F}$ is the matrix with learnable weights for the MHSA. $Concat()$ function [94; 76] combine h attention heads in the second axis: $\mathbb{R}^{L \times d} \to \mathbb{R}^{L \times (h \cdot d)}$.

Temporal MHSA

N T-MHSAs are employed to read in N locational sequences τ_n from the input encoder, and process temporal data for each node independently. Each T-MHSA will generate a new sequence for the corresponding node:

$$\tau_n^{new-T} = MHSA(Q_n, K_n, V_n) \tag{6.13}$$

where $\tau_n^{new-T} \in \mathbb{R}^{(4T-1)\times d}$; Q_n, K_n, V_n are projected from τ_n via corresponding transformation matrices as shown from (6.6)-(6.11). The last tokens $\tau_n^{new-T}(4T-1)$ in each sequence τ_n^{new-T} are weighted combinations of previous tokens and are organized into a sequence τ_T as the output of T-MHSAs:

$$\tau_T = (\tau_1^{new-T}(4T-1), \tau_2^{new-T}(4T-1), \cdots, \tau_N^{new-T}(4T-1)) \tag{6.14}$$

where $\tau_T \in \mathbb{R}^{N\times d}$. This sequence has N generated tokens for N nodes; each token $\tau_n^{new-T}(4T-1)$ only contains the temporal feature extracted from historical temporal data at node n. To share spatial information among different nodes, τ_T will be further processed by the following S-MHSA.

Spatial MHSA

S-MHSA reads in the sequence τ_T from T-MHSAs, and generates the following output to the decoder:

$$\tau^{new-S} = MHSA(Q_S, K_S, V_S) \tag{6.15}$$

where $\tau^{new-S} \in \mathbb{R}^{N\times d}$; Q_S, K_S, V_S are projected from τ_T via corresponding transformation matrices as shown from (6.7)-(6.11). This S-MHSA is designed to extract spatial correlations from the input sequence τ_T, which contains tokens generated for N different nodes across the market. For each token in this generated sequence, spatio-temporal correlated market data in all nodes at all time steps contribute differently. The spatio-temporal correlations among input sequential market data can be learned effectively and flexibly by the ST-MHSA network.

6.3.3 Output Decoder Neural Network

The decoder takes the sequence τ^{new-S} generated from ST-MHSA, which represents the hidden state parametrized by prior states of all nodes/locations and all time steps, to predict future action token in the sequential decision making process. The generated token is transformed to the forecasted RTLMPs for time step T via the projection of the output decoder, which is the inverse process of the input encoder:

$$LMP_{RT}^{T} = Decoder(\tau^{new-S}) \tag{6.16}$$

where $Decoder()$ is the learnable projection model built upon FC NNs.

In training, the learnable weights/models in the input encoder are learned to optimally represent the mapping between historical market data input and embedded sequence in (6.5); the learnable weights/models in ST-MHSA network are learned to optimally represent the spatio-temporal correlations among different tokens/data in embedded sequence in (6.6) which represent spatio-temporal correlations among different historical market data; the learnable weights/models in the output decoder are learned to optimally represent the mapping between the hidden state predicted by ST-MHSA network and the forecasted market price. Minibatches of sequences $(D_{DA}^{1}, D_{DA}^{2}, \cdots, D_{DA}^{T})$, $(D_{RT}^{1}, D_{RT}^{2}, \cdots, D_{RT}^{T})$, $(LMP_{DA}^{1}, LMP_{DA}^{2}, \cdots, LMP_{DA}^{T})$, $(LMP_{RT}^{1}, LMP_{RT}^{2}, \cdots, LMP_{RT}^{T-1})$ and $(FP^{1}, FP^{2}, \cdots, FP^{T})$ are sampled in each episode. The stacked decision transformer is trained to predict LMP_{RT}^{T} with mean-squared error. The overall structure of the proposed model and pseudocode are in Fig. 6.3 and Algorithm 2.

Algorithm 2 Stacked Decision Transformer Pseudocode

```
# D1, D2, L1, L2, F, t, n: DA demands, RT demands, DA LMPs, RT LMPs, fuel prices, time steps, nodes
# T: length of input sequence to Decision Transformer
# N: number of market nodes
# T-MHSA_n: T-MHSAs for each node
# S-MHSA: S-MHSA
# embed_D, embed_L, embed_F: linear embedding layers
# embed_t: learned timestep embedding
# pred_L: fully connected RTLMP prediction layer

# main model
def DecisionTransformer(D1,D2,L1,L2,F):
    # compute embeddings for tokens
    F_emb=embed_F(F)
    t_emb=embed_t(t) # per-timestep (not per-token)
    D1_embedding = embed_D(D1) + t_emb
    D2_embedding = embed_D(D2) + t_emb
    L1_embedding = embed_L(L1) + F_emb + t_emb
    L2_embedding = embed_L(L2) + F_emb + t_emb

    # interleave tokens as (D1_1, L1_1, D2_1, L2_2, ⋯, D1_T, L1_T, D2_T)
    input_embeds = stack(D1_embedding, L1_embedding,
                         D2_embedding, L2_embedding)

    # reorganize sequence to locational sequences
    for n in N:
```

```
        input_embeds_n=unstack(input_embeds).n

    # use T-MHSAs to get temporal hidden states
    for n in N:
        hidden_states_n = T-MHSA_n(input_embeds_n)

    # interleave hidden state tokens of T as (1,2,3, ··· N)
    hidden_states_N=stack(hidden_states_1_T, hidden_states_2_T,
                          hidden_states_3_T, ···, hidden_states_N_T)

    # use S-MHSAs to get spatio-temporal hidden states
    hidden_states_ST = S-MHSA(hidden_states_N)

    # predict RTLMP
    return pred_L(hidden_states_ST)

# training loop
for (D1, D2, L1, L2, F) in dataloader:
    L2_preds = DecisionTransformer (D1, D2, L1, L2, F)
    loss = mean((L2_preds-L2)**2)
    optimizer.SGD(); loss.backward(); optimizer.step()
```

6.4 Case Studies

The proposed RTLMP forecasting approach is tested using real-world public market data from SPP [4] and ISO-NE [3]. The proposed stacked decision transformer model is implemented with Tensorflow [76] and trained on Google Colab using online GPU for acceleration.

6.4.1 Performance Evaluation Metrics

In the case studies, the point-by-point LMP prediction accuracy is evaluated by the mean absolute percentage error (MAPE) [77]. The spatial correlation prediction is evaluated by the spatial prediction accuracy (SPA) and the spatial mean absolute percentage error (SMAPE) metrics [100]:

$$SPA(n) = 1 - \frac{\sum_{i=1}^{N} |sgn(Y_n - Y_i) - sgn(\hat{Y}_n - \hat{Y}_i)|/2}{N-1} \tag{6.17}$$

$$SMAPE(n) = \frac{\sum_{i=1}^{N} |(Y_n - Y_i) - (\hat{Y}_n - \hat{Y}_i)|}{\sum_{i=1}^{N} |(Y_n - Y_i)|} \tag{6.18}$$

where Y_i represents the ground truth RTLMP at node i ($i \neq n$); $\hat{Y}_i$ represents the predicted RTLMP at node i ($i \neq n$); $sgn(\cdot)$ is the sign function in (6.19),

$$sgn(z) = \begin{cases} -1 & if\ z<0 \\ 0 & if\ z=0 \\ 1 & if\ z>0 \end{cases} \tag{6.19}$$

SPA and SMAPE are adopted to quantify the spatial correlation prediction accuracy and error, respectively. Accurate prediction of spatial correlations among RTLMPs is expected to have higher SPA and lower SMAPE.

6.4.2 Test Case Description

The proposed spatio-temporal forecasting approach is implemented to forecast zonal and nodal RTLMPs in SPP and ISO-NE. For SPP market, historical market data of 16 price zones are collected; for ISO-NE, the historical market data of 9 price zones are collected. To make a fair comparison, a holistic prediction approach in [1] using the same SPP dataset is used as the evaluation benchmark. The componential and ensemble approach in [64], which consists of three individual extremely randomized tree (ET) regressors, is used as the evaluation benchmark for the ISO-NE case. The MAPEs obtained from the proposed approach are also compared with classical data-driven approaches, including standard NN, multivariate adaptive regression splines (MARS), pure ARMA model, and SVM. To demonstrate the proposed approach's capability of fast convergence and better spatial learning, the proposed approach is compared with our previous sequence prediction approach GAN, LSTM-GAN, and a naive DT NN.

Case 1

The RTLMP forecasting models are trained on SPP dataset, which contains hourly zonal DA demands, RT demands, DALMPs, and RTLMPs from June 2016 to July 2017, and historical natural gas price data. Additional historical generation mix data is incorporated in the benchmark models $ALG + \hat{M}$ proposed in [1] and a commercial price prediction product Genscape [1] (for comparison purposes). In Case 1(A), all models are tested to forecast hourly SPP RTLMPs of South Hub (Shub) and North Hub (Nhub) price nodes for 4 testing windows (07/31/2017-8/13/2017, 8/21/2017-9/3/2017, 9/18/2017-10/1/2017, and 10/2/2017-10/15/2017), in an hour-ahead manner. In Case 1(B), all models are tested to forecast hourly SPP RTLMPs in a day-ahead manner. The spatio-temporal decision transformer (ST-DT) model forecasts the next day's hourly RTLMPs step by step. In each step, the forecasted RTLMPs from the last time step are fed back into the ST-DT model along with RT demand forecast to conduct day-ahead forecast of RTLMPs.

Case 2

The RTLMP forecasting models are trained on ISO-NE dataset, which contains hourly zonal DA and RT market data from January 2019 to September 2019, and the corresponding historical natural gas price dataset. All models are tested to predict ISO-NE RTLMPs hour by hour from October 2019 to December 2019. All models are tested using input sequences with different lengths: (A) using past 12-hour historical data; (B) using past 24-hour historical data; (C) using past 48-hour historical data.

Case 3

The RTLMP forecasting models are trained on the SPP dataset to forecast nodal prices. The training dataset includes hourly nodal RTLMPs and DALMPs, hourly zonal DA demands and RT demands, and historical natural gas price data and generation mix data over two years (2016 and 2017). The models are tested to predict one year's SPP RTLMPs hour by hour from January 1, 2018, to December 31, 2018. All models are tested using an input sequence length of 24 hours. The training dataset includes 973 price nodes. In the testing, all models are tested to predict LMPs for 100 price nodes in SPP.

6.4.3 Model Construction and Configurations

For all decision transformers, the dimension of embedded input tokens is set to $M = 64$, $M = 32$, and $M = 128$ for case 1, case 2, and case 3, respectively; the projected queries Q, keys K and values V share the same dimension in each case; the numbers of attention heads $h(T)$ and $h(S)$ are set based on the input sequence length and number of market nodes. The setting details are listed in Table 6.1. In Table 6.1, parameters T, N, M, and d are total historical time steps, number of nodes/locations, dimension size of embedded input tokens, and dimension size of transformation matrices, respectively. All embedding layers are constructed with fully connected layers followed by batch normalization layers. The decoder consists of 2 layers of stacked fully connected networks. In the training process, the minibatch size is set to 8 for all models; a standard SGD optimizer is adopted.

Table 6.1: Spatio-Temporal Decision Transformer Setting

Case	T	N	M	d	$h(T)$	$h(S)$
1(A)	24	16	64	64	20	20
1(B)	24	16	64	64	20	20
2(A)	12	9	32	32	10	20
2(B)	24	9	32	32	20	20
2(C)	48	9	32	32	30	20
3	24	973	128	128	20	256

6.4.4 Performance Analysis

Case 1

The performance of the proposed ST-DT is compared with the standard fully connected NN, classic SA model, MARS, pure ARMA model, SVM model, GAN model, and CLSTM-GAN model.

Fig. 6.4 shows the ground-truth RTLMPs and forecasted RTLMPs at Shub price node over the testing period in Case 1(A). Fig. 6.4 demonstrates the proposed model's capability to capture the temporal dependencies of the ground-truth RTLMPs at Shub price node over the testing window. Table 6.2 lists and compares the mean absolute percentage errors (MAPEs) obtained using the proposed approach in Case 1(B) and other benchmark models. These approaches are tested using an identical SPP dataset. The ALG+$\hat{M}$ method proposed in [1] does not utilize DALMP data. Genscape [1] is a commercial product, which incorporates richer and proprietary confidential market data to forecast RTLMPs. Other benchmark models do not incorporate fuel price into

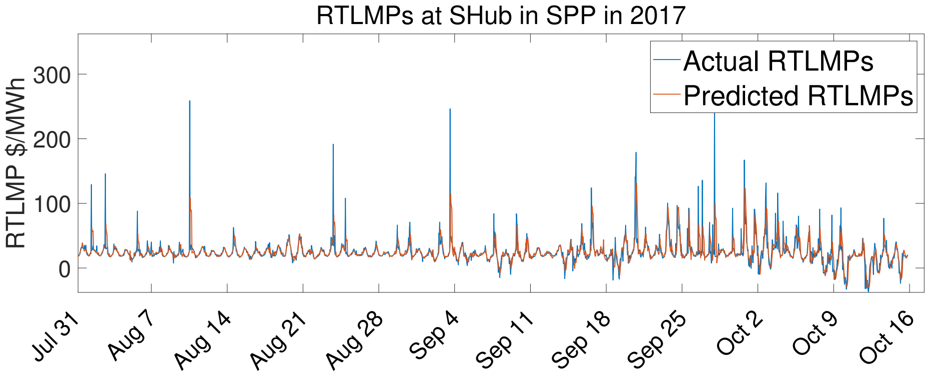


Figure 6.4: Ground-truth and Forecasted RTLMPs at SPP South Hub (SHub) Price Node in Case 1(A).

prediction. To make a fair comparison, an additional ST-DT is trained without using natural gas prices. The proposed approach outperforms the benchmark models in Case 1(B). Compared to state-of-the-art commercial RTLMP forecast product Genscape, the MAPEs of the proposed decision transformer model decrease by 10.1% and 25.2% for Shub and Nhub, respectively. Without using natural gas price information, the decision transformer can still achieve comparable performance to the industry benchmark Genscape, which uses confidential market data. Incorporating natural gas price data into the ST-DT approach can increase forecast accuracy significantly (decreasing MAPEs by 9.3% and 4.5% for Shub and Nhub, respectively).

Table 6.3 compares the spatial metrics SPA and SMAPE for the proposed ST-DT and several benchmark approaches in Case 1(A). Compared to the conventional SA and other sequential prediction approaches, which are not designed to capture spatial information, the proposed ST-DT has much better performance on spatial prediction. Our previous work GAN [100] is designed to learn spati-temporal correlated infor-

Table 6.2: RTLMP Forecasting Accuracy in Case 1 (B)

Approach	**MAPE(%) for Shub**	**MAPE(%) for Nhub**
NN	59.2	61.7
MARS	51.8	50.6
ARMA	37.4	39.2
SVM	40.9	40.8
SA	26.1	27.3
ALG+$\hat{M}$[1]	25.4	36.9
Genscape[2]	21.7	28.2
GAN	22.1	23.8
Case 1 (B)	19.5	21.1
Case 1 (B) without natural gas price	21.5	22.1

[1] The proposed method with the best performance in [1]

[2] State-of-the-art baseline prediction from Genscape[1]

mation in the electricity market. The proposed ST-DT approach has comparable performance to GAN. However, the GAN model requires a much longer training time and can easily diverge, while the proposed ST-DT approach has much less training time and much better convergence performance. The training times for SA, GAN, and ST-DT in Case 1 are 56 minutes, 583 minutes, and 71 minutes, respectively. The proposed ST-DT model can speed up the training significantly via parallel processing and obtain comparably high spatio-temporal prediction accuracy.

Table 6.3: Average SPA and SMAPE(%) in Case 1(A)

Model	NN	ARMA	SVM	SA	GAN	ST-DT
SPA	0.61	0.59	0.53	0.57	0.79	0.82
SMAPE	56	49	51	48	26	28

Case 2

Fig. 6.5 shows the ground-truth RTLMPs and forecasted RTLMPs at the ME price zone over one week in Case 2(B). The forecasted results in Fig. 6.5 indicate that the proposed ST-DT model successfully captures the temporal correlations of the ground-truth RTLMPs.

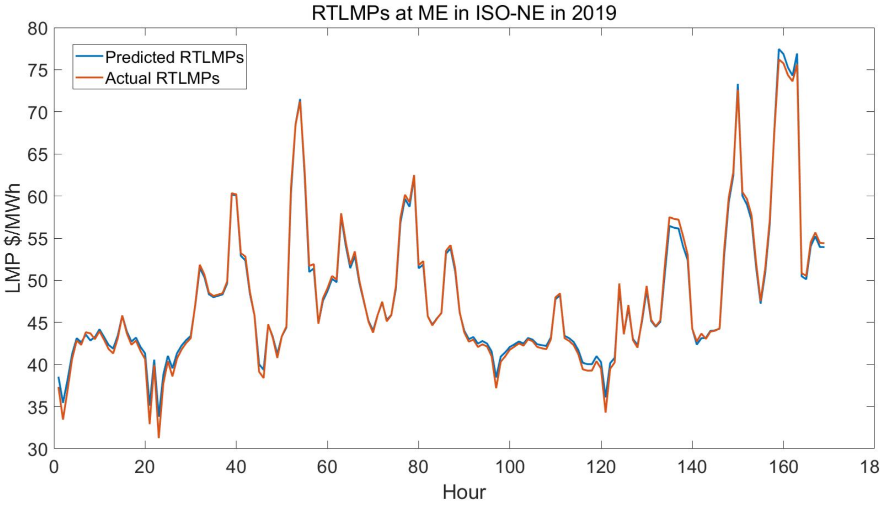


Figure 6.5: Ground-truth and Forecasted RTLMPs at ME Price Zone in ISO-NE in Case 2(B).

Table 6.4 compares next-hour forecasting MAPEs obtained for ISO-NE using different approaches on the same testing dataset in Case 2(A), 2(B), and 2(C). In Table 6.4, the ST-DT model outperforms all benchmark models. Compared to the MAPE in Case 2(A), increasing the length of the historical input sequence from 12 hours to 24 hours can improve the prediction accuracy by 16% for the decision transformer predictor. The CLSTM-GAN model has the second-best performance. Both the CLSTM-GAN model and the decision transformer model are sequence models. However, the CLSTM cells need to read in the time-series input data one by one. The decision transformer can read in and process the time-series input data parallelly. Therefore, the training time of the ST-DT model (59 minutes for Case 2(B)) is much less than the training time of CLSTM-GAN (614 minutes). Compared to the adversarial training procedure, the training procedure of the ST-DT model has better convergence performance.

Table 6.4: RTLMP Forecasting Accuracy in Case 2

Approach	**MAPE (%) in Case 2(A)**	**MAPE (%) in Case 2(B)**	**MAPE (%) in Case 2(C)**
NN	37.2	34.8	33.1
ARMA	30.7	25.1	26.3
SVM	27.7	25.2	25.6
CLSTM-GAN	15.4	12.09	13.1
ST-DT	11.9	10.3	10.1

Case 3

The long-time (one year) performance of the proposed ST-DT is compared with the standard fully connected NN, CLSTM model, classic SA model, MARS, and SVM model for different seasons. Table 6.5 lists and compares the MAPEs of different seasons (spring: March, April, May; summer: June, July, August; fall: September, October, November; winter: December, January, February) obtained using the proposed approach and other benchmark models. The proposed ST-DT outperforms all other benchmark models in Case 3. The results and comparison in Table 6.5 demonstrate ST-DT's capability to forecast LMPs for a large number of nodes in different seasons.

Table 6.5: RTLMP Forecasting Accuracy in Case 3

Approach	**MAPE (%) in spring**	**MAPE (%) in summer**	**MAPE (%) in fall**	**MAPE (%) in winter**
NN	35.3	36.7	32.5	33.8
CLSTM	27.8	28.1	26.7	28.0
MARS	30.9	30.7	31.5	31.3
SVM	25.1	26.3	25.2	26.7
SA	19.3	20.1	19.4	19.6
ST-DT	14.8	15.2	14.7	15.5

To demonstrate the ST-DT's spatial-aware capacity, Table 6.6 and Table 6.7 compare the spatial metrics SPA and SMAPE for the proposed ST-DT and other models. For all seasons, the proposed ST-DT has a much better performance on spatial prediction. CLSTM is designed to learn spatio-temporal correlated sequences and has the second-best performance. However, the CLSTM model requires a much longer training time and more parameters for a large system, while the proposed ST-DT approach has much less training time. The training times for CLSTM and ST-DT in Case 3 are 489 minutes, and 77 minutes, respectively. Compared to other spatial sequence learning approaches, the proposed ST-DT model can speed up the training significantly via parallel processing. Compared to the ST-DT model in Case 3, a larger ST-DT model to learn more nodes will not need much more training time because of the parallel processing.

Table 6.6: Average SPA in Case 3

Approach	**SPA in spring**	**SPA in summer**	**SPA in fall**	**SPA in winter**
NN	0.57	0.52	0.53	0.51
CLSTM	0.67	0.63	0.64	0.63
MARS	0.55	0.56	0.51	0.56
SVM	0.42	0.47	0.49	0.49
SA	0.53	0.57	0.53	0.56
ST-DT	0.84	0.87	0.86	0.81

Table 6.7: SMAPE(%) in Case 3

Approach	**SPA in spring**	**SPA in summer**	**SPA in fall**	**SPA in winter**
NN	40	42	41	42
CLSTM	36	37	35	40
MARS	47	49	44	46
SVM	49	42	43	45
SA	55	57	54	53
ST-DT	24	29	25	30

6.5 Conclusions and Future Work

This chapter proposes a stacked decision transformer-based approach for spatio-temporal forecast of RTLMPs, from market participants' perspective. The proposed approach performs well with different input horizons in online and day-ahead price forecasting problems for various market participants. The historical public market data are organized into a general decision marking sequence, which stores the spatio-temporal correlations of the market data. The RTLMP prediction problem is then formulated as a sequential decision-making problem and solved using the proposed spatio-temporal forecasting model ST-DT. The ST-DT consists of FC encoder, decoder, and ST-MHSA, which process the sequential data parallelly. Natural gas price data is embedded to improve prediction accuracy. Case studies using real-world historical market data from SPP and ISO-NE verify the performance of the proposed approach for both point-by-point price prediction accuracy and the accuracy of capturing spatial correlations among prices at different nodes. The proposed spatio-temporal forecasting approach can be directly applied to predicting both zonal and nodal LMPs.

Chapter 7

ENERGY PRICE PREDICTION CONSIDERING GENERATION BIDS VARIATION

LMP is composed of energy, congestion, and loss price components. All these price components are comprehensively determined by locational demands and locational generation bids. Due to difficulties in accessing updated generation bid information, previous price predictions from market participants' perspective focused only on learning the spatio-temporal correlations among historical price and load data without using generation bid information. The last chapter considered the generation bidding's effect on the market decision-making process. However, the historical bidding data published with delays can still not be utilized to improve LMP prediction accuracy. In this chapter, a two-stage CLSTM approach is proposed to incorporate historical generation bids into energy price prediction from market participants' perspective. Historical generation bids are organized into a 3D tensor and taken as the output of the first stage and input of the second stage in the training process. The implicit correlation among locational bids, demands, and energy prices is learned to improve price forecasting accuracy. Verification of the proposed approach is performed on the IEEE 30-bus system with publicly available historical market data from ISO-NE. Comparisons between the proposed approach and other state-of-art prediction approaches are conducted to demonstrate the improvement of the two-stage CLSTM approach.

7.1 Introduction

LMP is a common wholesale electricity pricing mechanism. As part of the standard market design promoted by the US Federal Energy Regulatory Commission (FERC), LMP at a node is defined as the cost increment for supplying one more MW demand at this node [9; 101]. As dual variables, LMPs are derived from solving an optimal power flow (OPF) problem by the system operator. The OPF formulation explicitly considers the locational demands, locational generation bids, and system network characteristics [101]. In most electricity markets, demand data and price data are always publicly available. System operating states and network characteristics are always confidential to market participants. Generation bidding data are usually published with several months' delay.

Without access to confidential network characteristics and latest generation bidding information, statistical methods and data-driven approaches using historical demand and price data are widely accepted to predict prices from market participants' perspective. Time-series-based statistical approaches, such as ARMAX model [37], and ARIMA model [30], are commonly utilized to learn correlations among historical LMPs. Correlations among locational demands, locational bids, and prices are ignored by these statistical approaches.

Assuming constant generation bidding, data-driven models are proposed to predict prices by learning SPRs [82]. In [83; 102], SPR-based approaches are developed to forecast prices using historical demand and price data, without considering generation bidding variations. However, different generation bids in real-world markets will lead to different SPRs. Fig. 7.1 shows 5 strategic bidding scenarios (10-block bids) of one generator from ISO-NE, which shows a generator's bids are not constant. Even within one operating day, a generator's hourly bids may vary significantly. Once generation bids change, identical load scenarios may point to different SPRs representing different prices. It is impractical to predict prices using SPR-based approaches (which ignore generation bidding variations) in real-world markets.

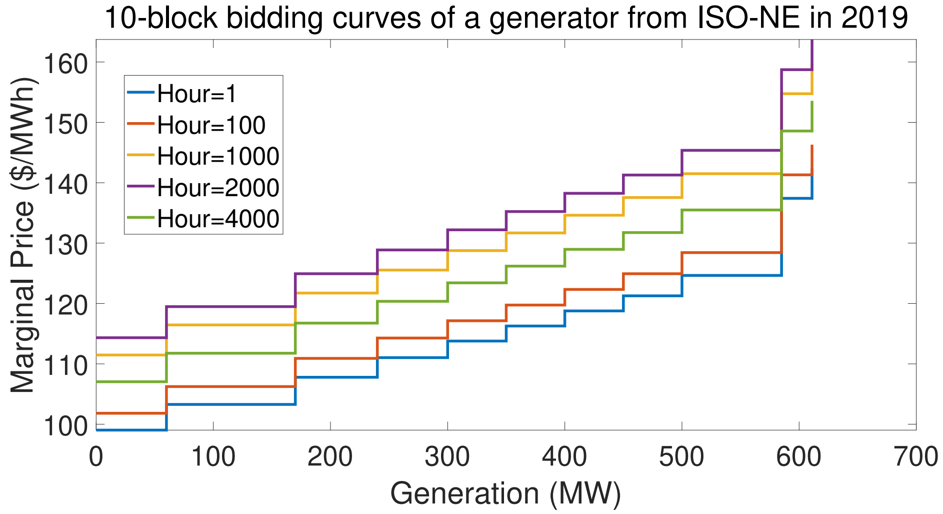


Figure 7.1: Bidding Curves of a Generator from ISO-NE in 2019.

With varying generation bids and demands, references [101] prove that all three components of LMP (energy, congestion, and loss prices) are jointly determined by the locational demands and locational generation bids. Even for the same total demand and same generation bids, the uncertainties of their distribution will affect the price components significantly.

The impact of locational demand and locational bids on the prices can be illustrated using the 6-bus test system built in Matpower. The 6-bus system has three generators (at Bus 1, 2, 3) and three loads (at Bus 4, 5, 6). All generators are allowed to submit 5-block bids for every hour. For the basic bidding strategy, the quantity offer for each block is 20 MW; the price offers of five blocks are 10, 15, 20, 25, 30 \$/MWh.

Table 7.1: Price Components ($/MWh) in 6-Bus Example

Case No.	**Bus 1**	**Bus 2**	**Bus 3**	**Bus 4**	**Bus 5**	**Bus 6**
	Energy price	**Congestion price**				
1)	25	-0.91	-0.63	1.11	-0.30	-0.65
2)	29.47	0.58	-27.89	0.17	-1.13	25.92
3)	35	-22.71	-15.71	27.64	-7.38	-16.10

Consider the following 3 scenarios (Scenarios 2 and 3 are modified based on Scenario 1): 1) each load is 70 MW, and each generator has the same basic bidding strategy; 2) loads at Bus 4 and Bus 5 are 60 MW, and load at Bus 6 is 90 MW; 3) generator at Bus 1 increases the price offers of the 4th block and 5th block to 35 $/MWh and 45 $/MWh, respectively.

In a system, the energy price components for all nodes are identical [9; 101]; the loss price components are usually much lower compared with the energy and congestion price components. Table 7.1 shows the energy price at Bus 1 (reference bus, congestion prices are zero) and congestion prices at all the other buses. Table 7.1 shows a slight variation of either the load distribution or bidding strategy can cause significant variation in LMP components. Therefore, using partial market data without generation bids or improper assumption of fixed bidding behavior is a main source of price prediction errors.

Chapter 4 and Chapter 5 studied the spatio-temporal correlations among historical LMPs, using the latest public market data. Chapter 5 shows the CLSTM network's capability to capture spatial correlations among LMPs, which are highly related to predicting congestion prices. To further improve price prediction accuracy, this chapter proposes a two-stage CLSTM approach to learn the implicit correlation among energy prices, locational demands, and locational generation bids using historical generation bid data (with several months' publication delay) during training.

The rest of this chapter is organized as follows. Section II introduces the data structure for locational market data; Section III proposes the two-stage CLSTM model for energy price prediction; Section IV verifies the performance of the proposed approach using a 30-bus system and real market data from ISO-NE; Section V concludes this chapter.

7.2 Market Data Structure

In this section, the general 3-dimensional (3D) data structure proposed in Chapter 4.1 is directly applied to organize historical demand data and price data. The historical monotonically increasing step-wise bid curves (such as Fig. 7.1) are also transformed into a similar 3D tensor structure.

7.2.1 Multi-Channel 3D Tensor of Demands and LMPs

Consider a set of publicly available locational demand data and LMP data collected from $N = m \times n$ price nodes for T consecutive hours. According to the historical market data tensor definition, the historical data set of locational demand and LMP is organized into a two-channel 3D tensor $\mathcal{X} \in \mathbb{R}^{m \times n \times T}$. Each element $x_{i,j}^{t}$ in this tensor $\mathcal{X}$ contains two channels storing demand data and LMP data. $\mathcal{X}$ consists of a sequence of 2D arrays, $\mathcal{X} = \{X^1, X^2, \cdots, X^T\}$.

7.2.2 Multi-Channel 3D Tensor of Generation Bids

Consider a set of locational monotonically increasing step-wise generation bid curve data collected from $M = h \times w$ different price nodes at one hour. Each generator is allowed to submit b blocks of bids. Let $P_k = \{p_k^1, p_k^2, \cdots, p_k^b\}$ be generation prices (in \$/MWh) of a generator's bid, where $k \in [1, M]$, $p_k^1 \leq p_k^2 \leq \cdots \leq p_k^b$. Let $Q_k = \{q_k^1, q_k^2, \cdots, q_k^b\}$ be generation quantities (in MW) of a generator's bid. The locational generation bid data P_k and Q_k are published by the system operator with delay and can be organized into a 3D tensor $\mathcal{Y} \in \mathbb{R}^{h \times w \times b}$ as shown in Fig. 7.2.

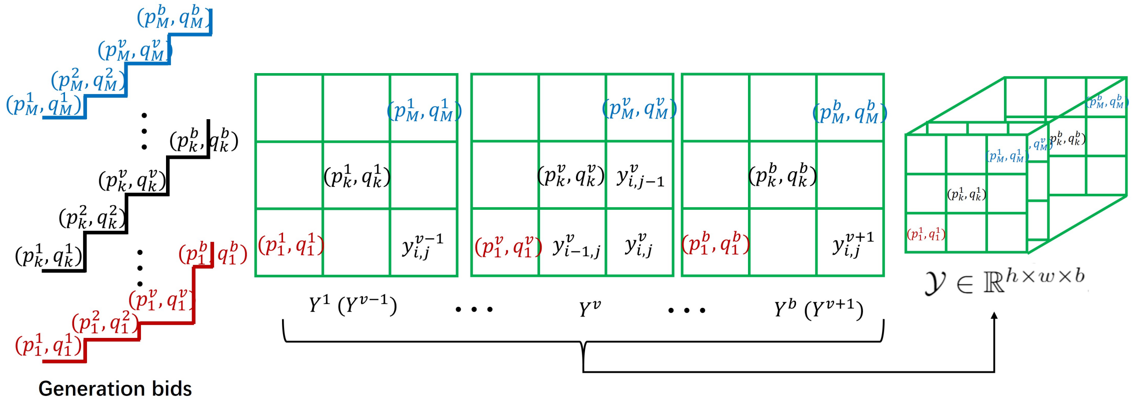


Figure 7.2: Generation Bids Data Structure.

Each element $y_{i,j}^v$ in tensor $\mathcal{Y}$, contains two channels storing price and quantity data and represents v^{th} generation bid block of k^{th} generator ($y_{i,j}^v = (p_k^v, q_k^v)$), where $v \in [1, b]$; $k = h \times (i - 1) + j$. $\mathcal{Y}$ consists of a sequence of 2D arrays, $\mathcal{Y} = \{Y^1, Y^2, \cdots, Y^b\}$. The correlations among different positions in one 2D array Y^v, such as $y_{i-1,j}^v$, $y_{i,j}^v$ and $y_{i,j-1}^v$ in Fig. 7.2, represent the bidding differences among different generators. The correlations among the same positions at different 2D arrays, such as $y_{i,j}^{v-1}$, $y_{i,j}^v$ and $y_{i,j}^{v+1}$ in Fig. 7.2, represent bidding strategy of one generator. The correlations among the same positions at different 3D tensors represent the bidding strategy differences of one generator at different hours.

7.3 Two-Stage CLSTM Network

Even though published with several months' delay, bidding characteristics can still be learned and leveraged to improve price prediction accuracy. This section proposes a two-stage CLSTM approach that takes advantage of historical bid data (not up-to-date due to publication delay) to estimate generation bidding strategies in offline training and utilizes the trained prediction model with learned generation bidding strategies to predict energy prices online with the latest public market data.

7.3.1 Formulation of Energy Price Prediction

From the market participants' perspective, the traditional objective of energy price prediction is to use available inputs of the OPF model to estimate the outputs without knowing the characteristics of the OPF model. Let $\mathcal{X}$ represent available historical demand and price data; Let $\hat{Z}$ and Z denote the predicted energy price and the ground truth at time step $T+1$, respectively. The well-trained predictor should generate the most likely future price given historical data $\mathcal{X}$, which can maximize conditional probability as:

$$\hat{Z} = \arg\max_{\tilde{Z}} p(\tilde{Z} = Z|\mathcal{X}) \tag{7.1}$$

However, the conditional probability in (7.1) cannot reflect the actual OPF problem. Let $\mathcal{Y}(T+1)$ in Section II.B denote the ground-truth generation bid data. According to the formulation of OPF, the predicted price should maximize the following joint conditional probability:

$$\hat{Z} = \arg\max_{\tilde{Z}} p(\tilde{Z} = Z|X^{T+1}, \mathcal{Y}(T+1)) \tag{7.2}$$

For example, let a market participant predict the next hour's prices. The OPF model is always implicit to the market participant. A well-trained neural network is desired to mimic the OPF process to maximize conditional probability in (7.2), working as a black box. This pure data-driven predictor does not require any confidential model information or explore any specific OPF constraints, which are not available to market participants. However, system operators can always incorporate OPF constraints as needed to improve prediction accuracy further. In our proposed approach, a CLSTM(S2) network is employed to estimate prices based on demands and bids. According to the market timeline in Fig. 7.3, the latest generation bid information $\mathcal{Y}(T+1)$ is unknown to market participants in actual prediction (due to publication delay). Therefore, another well-trained neural network is desired to guess the potential bids based on historical data. A CLSTM(S1) network is employed to estimate $\hat{\mathcal{Y}}(T+1)$ by learning historical bidding characteristics and the correlations among available historical market data. The most likely bidding strategy estimated by CLSTM(S1) should maximize following conditional probability, where $\mathcal{Y}(T+1)$ denotes the ground truth of $\hat{\mathcal{Y}}(T+1)$:

$$\hat{\mathcal{Y}}(T+1) = \underset{\tilde{\mathcal{Y}}(T+1)}{\arg\max}\, p(\tilde{\mathcal{Y}}(T+1) = \mathcal{Y}(T+1)|\mathcal{X}) \tag{7.3}$$

Substituting (7.3) into (7.2), the joint conditional probability maximized in (7.2)

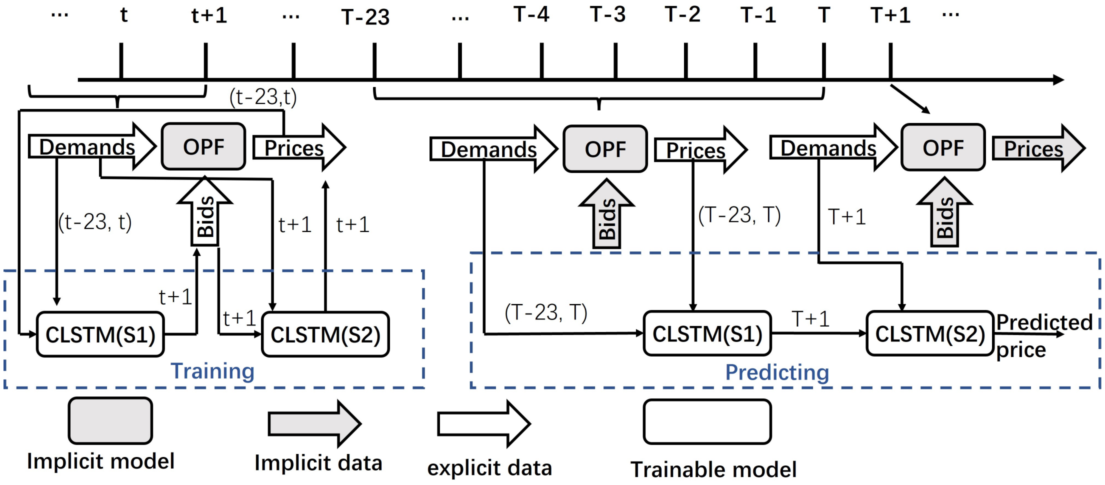


Figure 7.3: Market Timeline and the Overall Framework of Two-stage CLSTM Prediction Approach.

should be represented as:

$$p(\tilde{Z}|X^{T+1},\mathcal{X}) = p(\tilde{Z}|X^{T+1},\hat{\mathcal{Y}}(T+1))p(\hat{\mathcal{Y}}(T+1)|\mathcal{X}) \tag{7.4}$$

As shown in Fig. 7.3, during training, all historical market data including demands, bids, and prices are explicit. The CLSTM(S1) is trained to generate the most likely estimation of bidding strategy satisfying (7.3); the 2nd-stage network CLSTM(S2) is trained to generate the most likely prediction of energy price satisfying (7.2). In actual prediction, the latest historical demands and prices are explicit; the latest historical bids are implicit to market participants. The CLSTM(S1) takes the past day's demand and price data as inputs, and generates an estimated bidding strategy as output; CLSTM(S2) takes the bidding strategy estimation from the first stage (output of CLSTM(S1)) and the demand data as inputs to predict future prices, which maximize the joint conditional probability in (7.4). This offline training approach allows the predictor to extract bidding strategies from historical generation bidding data. These learned strategies are utilized in online price prediction to substitute the latest bidding data which is unavailable.

7.3.2 Convolutional Long Short-Term Memory Network

The CLSTM cell in [89] is adopted for CLSTM(S1); The CLSTM cell is modified to a conditional cell for CLSTM(S2). Three advantages make CLSTM and conditional CLSTM networks ideal for electricity price prediction: 1) LSTM structure is designed to learn temporal correlations for sequence forecasting with sequence data inputs, and the input market data structures in our price prediction are sequences of 2D arrays as shown in Section II; 2) both demand and generation bid data are locational data whose spatial correlations should be learned, and the CLSTM network replaces fully connected structures in classical LSTM networks (effective for learning temporal characteristics) with convolution operators (effective for learning spatial characteristics), allowing CLSTM to capture spatio-temporal correlations in historical data; 3) to mimic OPF process, CLSTM(S2) should take a sequence of bid data conditioned on one hour's locational demands, which falls within the general formulation of conditional CLSTM.

As shown in Fig. 7.4, the CLSTM(S1) runs 24 times to read the past day's market data $\mathcal{X} = \{X^{T-23}, X^{T-22}, \cdots, X^{T-1}, X^{T}\}$. The output of CLSTM(S1) is the estimated generation bidding strategy $\hat{\mathcal{Y}}(T+1) = \{\hat{Y}^1, \hat{Y}^2, \cdots, \hat{Y}^b\}$. The internal structure details of the CLSTM cell are in [89]. CLSTM(S2) contains conditional CLSTM cell as shown in Fig. 7.5 and fully connected layers. The conditional CLSTM cell runs b times to read bidding strategy $\hat{\mathcal{Y}}(T+1)$ or $\mathcal{Y}(T+1)$ conditioned on the locational demands (demand forecast) X^{T+1} (X^{T+1} is a single channel 2D array storing only demand data). The output of CLSTM(S2) is the predicted energy price. In each running step t ($t \geq 2$), conditional CLSTM cell performs following operation:

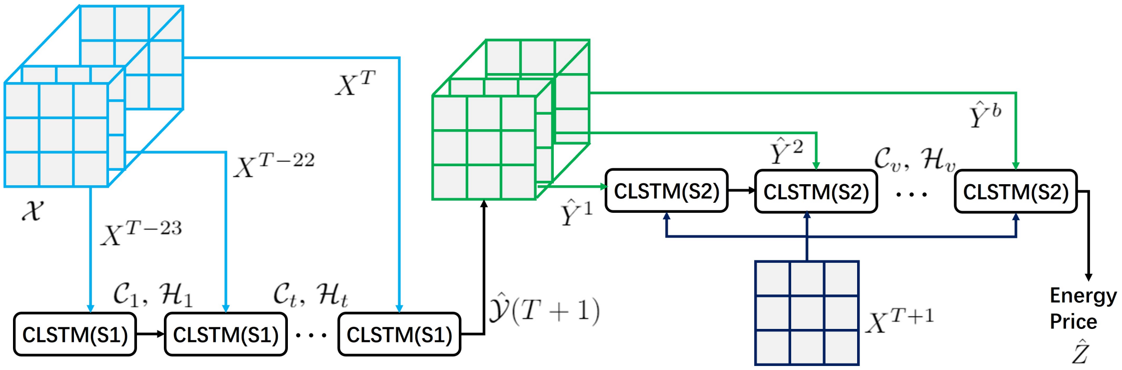


Figure 7.4: The Structure of Two-stage CLSTM Predictor.

$$i_t = \sigma(W_{xi} * Y^t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + W_{bi} * X^{T+1}) \quad (7.5)$$

$$f_t = \sigma(W_{xf} * Y^t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + W_{bf} * X^{T+1}) \quad (7.6)$$

$$o_t = \sigma(W_{xo} * Y^t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_{t-1} + W_{bo} * X^{T+1}) \quad (7.7)$$

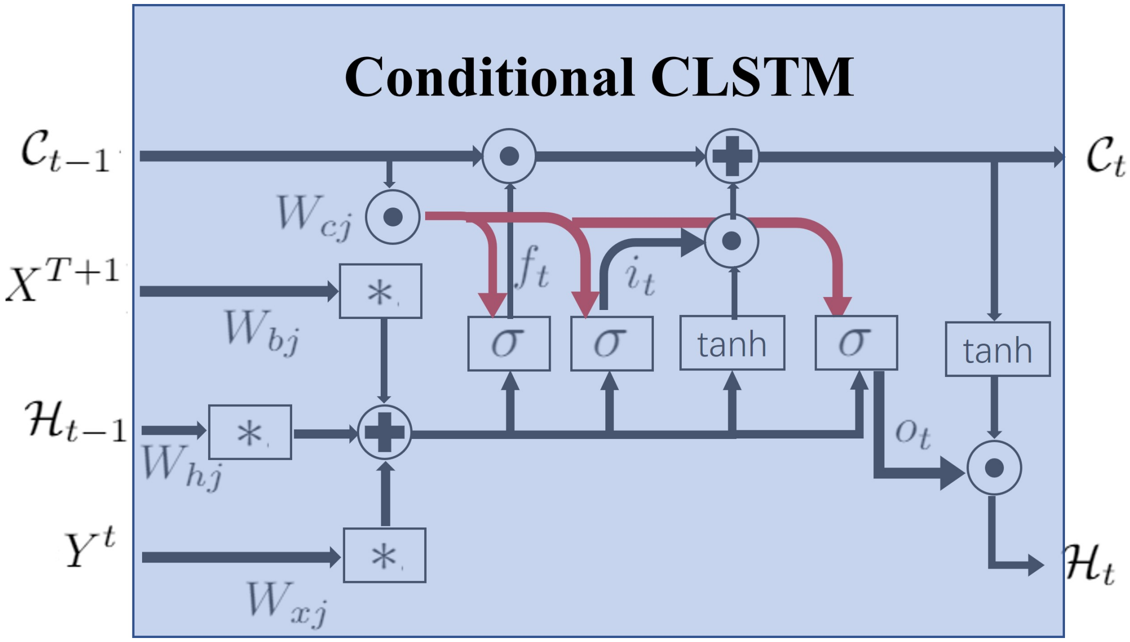


Figure 7.5: Conditional CLSTM Cell.

$$\mathcal{C}_t = f_t \circ \mathcal{C}_{t-1} + i_t \circ tanh(W_{xc} * Y^t + W_{hc} * \mathcal{H}_{t-1} + W_{bc} * X^{T+1}) \tag{7.8}$$

$$\mathcal{H}_t = o_t \circ tanh(\mathcal{C}_t) \tag{7.9}$$

where $*$, $\circ$, $\sigma(\cdot)$ denotes the convolution operator, Hadamard product, and sigmoid activation function, respectively; i_t, f_t, o_t are internal gates for CLSTM cell; the weight matrices W_{xj}, W_{hj}, W_{bj} (with $j \in \{i, f, c, o\}$), W_{cj} (with $j \in \{i, f, o\}$) are learnable parameters to be optimized in training. As shown in Figure **??**, in the $v^{th} (v \geq 2)$ step, CLSTM(S2) takes three sets of inputs: 1) the 2D estimated locational bids of v^{th} block, $\hat{Y}^v$; 2) the hidden state array $\mathcal{H}_{v-1}$ and output array $\mathcal{C}_{v-1}$ from previous step; 3) locational demand data. The convolution operations $W_{bj} * X^{T+1}$ (with $j \in \{i, f, c, o\}$) in (7.5)-(7.8) make CLSTM(S2) conditioned on the demand data in each step.

7.3.3 Loss Functions

During batch training, learnable parameters in CLSTM(S2) are optimized by minimizing mean absolute percentage error (MAPE) [103] as $MAPE = \frac{1}{M}\sum_{i=1}^{M}(\left|\frac{P_i - \hat{P}_i}{P_i}\right|)$, where M is the batch size, $\hat{P}_i$ denotes the predicted price; P_i denotes the ground truth. The CLSTM(S1) is trained through minimizing the following loss function: $\mathcal{L}_2(\hat{\mathcal{Y}}, \mathcal{Y}) = \frac{1}{b}\sum_{i=1}^{b}\left\|\hat{Y}^i - Y^i\right\|_2^2$, where $\|\cdot\|_2$ denotes the entry-wise 2-norm of a 2D array. The selection of loss function in training will significantly affect the final prediction accuracy of spatial-temporal correlated data. In Chapter 4.1, a multi-loss function is designed for the learning of spatial-temporal correlated data, and will be applied to the proposed model in future work.

7.4 Case Study

The proposed model is verified on an IEEE 30-bus system. To illustrate the proposed approach's capability to learn the correlations among prices, locational demands, and locational generation bids in the real-world market, the training and testing data are mapped from historical market data in ISO-NE.

7.4.1 30-bus System

System Configuration

Most of the system settings, such as bus and branch parameters, load locations, generator locations, and limits, follow the 30-bus system in [104].

Generation

6 generators are located at 6 buses, with a total generation capacity of 335MW. Since ISO-NE allows each generator to submit hourly 10-block generation bids, the

generation costs in this system are set as 10-step piece-wise linear functions, linearly mapped from historical bid data of 6 generators from ISO-NE real-time market in 2019.

Load

In this system, there are 20 fixed loads, with a total system load of 189.2 MW in the benchmark load profile. An hourly locational load profile including 20 locational demands from ISO-NE real-time and day-ahead markets in 2019, is used to map the benchmark load profile to a one-year hourly real-time locational load profile as follows,

$$\tilde{L}_i(t) = \frac{L_i(t)}{L_T(t)}[0.5 + (\frac{L_T(t) - L_{min}}{L_{max} - L_{min}})(1.5 - 0.5)] \times 189.2MW \tag{7.10}$$

where $\tilde{L}_i(t)$ (with $i \in [1, 20]$, $t \in [1, 8760]$) denotes the i^{th} load at hour t of the 30-bus system; $L_i(t)$ denotes the i^{th} load at hour t collected from ISO-NE; $L_T(t) = \sum_{i=1}^{20} L_i(t)$; $L_{min} = min(L_T(t)), \forall t \in [1, 8760]$; $L_{max} = max(L_T(t)), \forall t \in [1, 8760]$.

OPF Simulation

With hourly real-time generation cost and load profile, Matpower solves OPF to derive hourly real-time LMPs at all buses. LMPs are decomposed into energy price, congestion price, and loss price components based on reference Bus 1.

7.4.2 Test Case Description

Historical generation bids, locational demands, and corresponding prices are divided into training and testing data sets by a ratio of 8 : 2. In training, CLSTM(S1) takes historical 2-channel 3D tensor $\mathcal{X} \in \mathbb{R}^{4\times5\times24}$ of past 24-hour market data to estimate next-hour generation bids $\hat{\mathcal{Y}} \in \mathbb{R}^{2\times3\times10}$; CLSTM(S2) takes 3D tensor $\mathcal{Y} \in \mathbb{R}^{2\times3\times10}$ conditioned on 2D array of locational demands $X \in \mathbb{R}^{4\times5}$ to predict the next-hour energy price. In testing, CLSTM(S2) takes $\hat{\mathcal{Y}} \in \mathbb{R}^{2\times3\times10}$ estimated by CLSTM(S1) to predict future energy price. The classic CLSTM model and other popular data-driven approaches, such as standard fully connected neural network (FCNN), SVM, and SDA [77], are implemented as benchmark models. Because all these benchmark models are designed to predict LMPs without considering incorporating historical generation bid data, they only take historical prices and locational demands as inputs. To make a fair comparison, a two-stage fully connected neural network (2SFCNN) is implemented. The 2SFCNN model replaces the CLSTM structures in the proposed approach with two fully connected neural networks. The 2SFCNN and our proposed model take identical training inputs including historical generation bids.

7.4.3 Neural Network Architecture and Configurations

The proposed model is implemented with Tensorflow 2.0 [76] and trained using online GPU on Google Colaboratory [105]. The architecture details of CLSTM(S1) and CLSTM(S2) are listed in Table 7.2, where 'ConvLSTM2D' denotes the convolutional layer with CLSTM cells; 'Conv3DTranspose' denotes the convolutional transpose layer; 'Conv3D' denotes the convolutional layer; 'Flatten' denotes the operation of flatting the input, and 'Dense' denotes the fully connected layer. The 'ConvLSTM2D' layers in CLSTM(S2) are customized with a standard convolutional layer with CLSTM cells, and extra convolutional operations, such that the conditional CLSTM operations in (7.5)-(7.9) are implemented. All the 'ConvLSTM2D', 'Conv3D', and 'Conv3DTranspose' layers are followed by batch normalization layers and ReLU units. All 'ConvLSTM2D' layers accept $channel - last$ data format and *sigmoid* recurrent activation functions. The standard stochastic gradient descent (SGD) optimizer is utilized for the training of CLSTM(S1) and CLSTM(S2). In the

Table 7.2: Neural Network Architecture Details

Layers	**CLSTM(S1)** (Layer Type, Feature Map)	**CLSTM(S2)** (Layer Type, Feature Map)
Input	$24 \times 4 \times 5$	$2 \times 3 \times 10$, 4×5
Layer 1	ConvLSTM2D, 128	ConvLSTM2D, 128
Layer 2	ConvLSTM2D, 256	Conv3DTranspose, 256
Layer 3	Conv3D, 128	Flatten
Layer 4	Conv3DTranspose, 64	Dense, 512
Layer 5	Conv3DTranspose, 10	Dense, 1
Output	$2 \times 3 \times 10$	Predicted energy price

training process, the minibatch size is 4 for both CLSTM(S1) and CLSTM(S2).

7.4.4 Case Study Results

The proposed two-stage CLSTM model and all other benchmark models are tested on the same data set including only historical demand and price data (without the delayed bidding data). Fig. 7.6 shows hourly energy prices predicted by the two-stage CLSTM model and the ground truth during one week. The predicted energy prices successfully follow the actual prices. The proposed approach is able to accurately estimate generator bidding behavior using historical market data and mimic the OPF solutions. However, there is still an obvious mismatch of the price spike in Fig. 7.6. This is caused by an estimation mismatch of unusual bidding strategies. In operating days with contingencies or extreme weather, market participants intend to arbitrage by manipulating real-time generation bids, which are difficult to estimate using historical data.

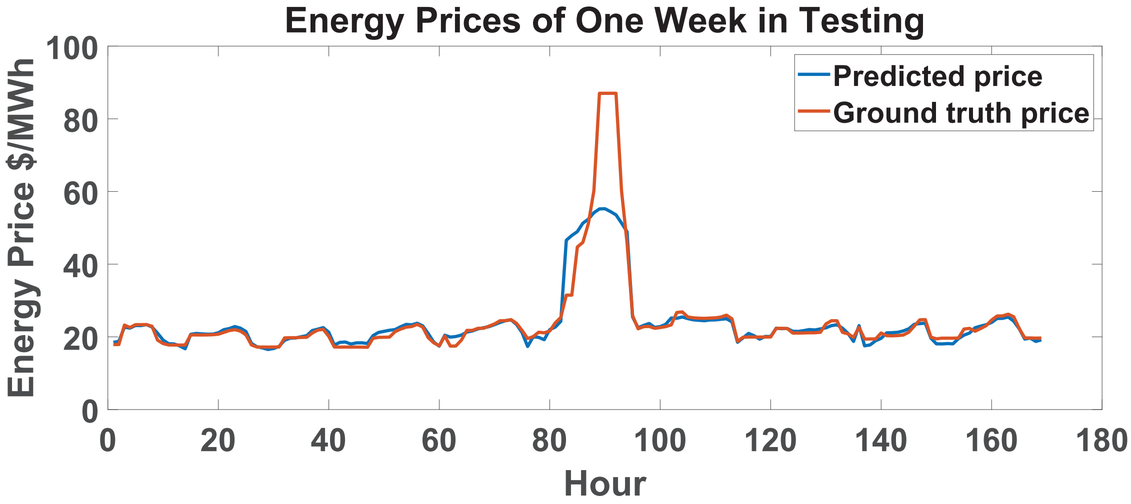


Figure 7.6: Ground-truth and Predicted Energy Price of One Week.

The MAPEs of all tested models are compared in Table 7.3. The proposed model (with a MAPE of 8.03%) outperforms the state-of-the-art SDA forecasting model (with a MAPE of 11.15%) and all other benchmark models. Utilizing historical bid data (with publication delay) in training, the two-stage CLSTM model can improve prediction accuracy significantly. Compared with the FCNN model without using historical generation bid data in training, the 2SFCNN model achieves limited improvement (6.9%) by incorporating historical generation bid data. Following the same workflow, the proposed model improves accuracy by 39.7% compared to the MAPE of 2SFCNN, which demonstrates the conditional CLSTM's capability to learn sequential locational market data. The prediction accuracy of CLSTM(S2) depends on the accuracy of bidding strategy estimation from CLSTM(S1). Let CLSTM(S2) predict energy prices using ground-truth bids, instead of estimations made by CLSTM(S1), the prediction accuracy is 6.09% (improved by 24.16% compared to MAPE of 8.03% in Table 7.3).

Table 7.3: Energy Price Prediction Errors

Model	Classic CLSTM	FCNN	SVM	SDA	2SFCNN	Two-stage CLSTM
MAPE(%)	12.23	14.29	15.61	11.15	13.31	8.03

7.5 Conclusions

This chapter proposes a two-stage CLSTM approach to predict energy prices from market participants' perspective. This approach takes advantage of historical generation bid data published with months' delay in the training process to learn implicit correlations among prices, demands, and bids. The well-trained model predicts energy prices accurately using only the latest available market data. In the case study using the IEEE 30-bus system and ISO-NE's historical market data, the proposed model successfully captures weekly energy price characteristics and outperforms other approaches.

Chapter 8

CONCLUSION AND FUTURE WORK

This dissertation proposes a general 3D tensor structure to organize system-wide heterogeneous market data streams, which store the spatio-temporal correlations. This general data structure is flexible for prediction problems with different time horizons. Data-driven LMP prediction approaches are developed from the market participants' perspective. A GAN-based approach is proposed to predict system-wide LMPs. The RTLMP prediction problem is formulated as a 2D array prediction problem and solved using the proposed deep convolutional GAN model with multiple loss functions. The prediction accuracy is improved by an ARMA calibration approach to mitigate deviations caused by variation/uncertainty of generation bids. To further improve forecasting accuracy, a CLSTM-based GAN is proposed to forecast LMPs from market participants' perspective. The LMP forecasting problem is formulated as a spatio-temporal sequence-to-sequence forecasting problem. Considering the generation bidding's effect on the market clearing process, an ST-DT approach is proposed to learn the sequential decision-making and forecast RTLMPs. To incorporate historical generation bids into price prediction from market participants' perspective, a two-stage CLSTM approach is proposed. Historical generation bids are organized into a 3D tensor and taken as the output of the first stage and input of the second stage in the training process. The implicit correlation among locational bids, demands, and energy prices is learned to improve price forecasting accuracy. Case studies using real-world historical market data from ISO-NE, SPP, and MISO verify the performance of the proposed data-driven approaches for both point-by-point price prediction accuracy and accuracy of capturing spatial correlations among prices at different locations.

Even though the proposed prediction approaches have comparable performance compared to the state-of-the-art industrial benchmark, and have significant improvement compared to the existing prediction methods which also use public market data only, the prediction error should be reduced further. As shown in the study cases, several spikes are missed, which contribute most to the prediction errors. Future work could focus on price spikes prediction by incorporating additional public contingency data and market participants' strategic behaviors. One potential area of exploration includes predicting the energy price component of LMP by learning the total supply-demand balance. The congestion price component and loss price component may be predicted by other spatial learning methods. The final spatio-temporal correlated nodal LMPs can be assembled based on these three individual price components. Because the LMPs are a consequence of generators' strategic bidding behaviors, which are considered the result of a decision-making process based on the knowledge of historical market data and current fuel prices, another potential area of exploration is to construct the two-stage approach using two decision transformers. The first stage decision transformer could be responsible for the generator's strategic bidding behavior learning. The second stage decision transformer could be responsible for learning the approximation of an optimization problem.

REFERENCES

[1] A. Radovanovic, T. Nesti, and B. Chen, "A holistic approach to forecasting wholesale energy market prices," *IEEE Transactions on Power Systems*, pp. 1–1, 2019.

[2] "Pjm interregional data map," 2020. [Online]. Available: https://pjm.com/markets-and-operations/interregional-map.aspx

[3] "Iso new england real-time market data," 2020. [Online]. Available: https://www.iso-ne.com/isoexpress/

[4] "Spp data," 2020. [Online]. Available: https://marketplace.spp.org/

[5] "Miso market data," 2022. [Online]. Available: https://api.misoenergy.org/MISORTWD/lmpcontourmap.html

[6] "Basics of the classic cnn," 2019. [Online]. Available: https://towardsdatascience.com/basics-of-the-classic-cnn-a3dce1225add

[7] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International Journal of Forecasting*, vol. 30, no. 4, pp. 1030–1081, 2014.

[8] "Energy primer," 2015. [Online]. Available: https://www.ferc.gov/market-assessments/guide/energy-primer.pdf

[9] ISO New England, "Market rule 1," 2019. [Online]. Available: www.iso-ne.com/participate/rules-procedures/tariff/market-rule-1

[10] F. C. Schweppe, *Spot pricing of electricity / by Fred C. Schweppe ... [et al.].* Kluwer Academic Boston, 1988.

[11] V. Kekatos, G. B. Giannakis, and R. Baldick, "Online energy price matrix factorization for power grid topology tracking," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1239–1248, May 2016.

[12] Q. Zhou, L. Tesfatsion, and C. Liu, "Short-term congestion forecasting in wholesale power markets," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2185–2196, 2011.

[13] G. Hamoud and I. Bradley, "Assessment of transmission congestion cost and locational marginal pricing in a competitive electricity market," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 769–775, May 2004.

[14] J. Bastian, Jinxiang Zhu, V. Banunarayanan, and R. Mukerji, "Forecasting energy prices in a competitive market," *IEEE Computer Applications in Power*, vol. 12, no. 3, pp. 40–45, July 1999.

[15] R. Deb, R. Albert, L.-L. Hsue, and N. Brown, "How to incorporate volatility and risk in electricity price forecasting," *The Electricity Journal*, vol. 13, no. 4, pp. 65 – 75, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1040619000001020

[16] Y. Ji, R. J. Thomas, and L. Tong, "Probabilistic forecast of real-time lmp via multiparametric programming," in *2015 48th Hawaii International Conference on System Sciences*, 2015, pp. 2549–2556.

[17] W. Deng, Y. Ji, and L. Tong, "Probabilistic forecasting and simulation of electricity markets via online dictionary learning," 2016.

[18] P. R. Kleindorfer, D.-J. Wu, and C. S. Fernando, "Strategic gaming in electric power markets," *European Journal of Operational Research*, vol. 130, no. 1, pp. 156 – 168, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221700000485

[19] B. E. Hobbs, "Linear complementarity models of nash-cournot competition in bilateral and poolco power markets," *IEEE Transactions on Power Systems*, vol. 16, no. 2, pp. 194–202, 2001.

[20] C. Metzler, B. F. Hobbs, and P. Jong-Shi, "Nash-cournot equilibria in power markets on a linearized dc network with arbitrage: Formulations and properties," *Networks and Spatial Economics*, vol. 3, no. 2, p. 123, 06 2003, copyright - Copyright Kluwer Academic Publishers Jun 2003; Last updated - 2014-08-30. [Online]. Available: http://login.ezproxy1.lib.asu.edu/login?url=https://search-proquest-com.ezproxy1.lib.asu.edu/docview/205850147?accountid=4485

[21] C. M. Ruibal and M. Mazumdar, "Forecasting the mean and the variance of electricity prices in deregulated markets," *IEEE Transactions on Power Systems*, vol. 23, no. 1, pp. 25–32, 2008.

[22] F. Nogales, J. Contreras, A. Conejo, and R. Espinola, "Forecasting next-day electricity prices by time series models," *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 342–348, 2002.

[23] M. Shahidehpour, *Market operations in electric power systems : forecasting, scheduling, and risk management.* New York]: Institute of Electrical and Electronics Engineers, Wiley-Interscience, 2002.

[24] R. Weron, *Modeling and forecasting electricity loads and prices : a statistical approach*, ser. Wiley finance series. Chichester, England ; Hoboken, NJ: John Wiley Sons, 2006.

[25] T. Jonsson, P. Pinson, H. A. Nielsen, H. Madsen, and T. S. Nielsen, "Forecasting electricity spot prices accounting for wind power predictions," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 1, pp. 210–218, 2013.

[26] J. S. McMenamin and F. A. Monforte, *Statistical Approaches to Electricity Price Forecasting.* Boston, MA: Springer US, 2000, pp. 249–263. [Online]. Available: https://doi.org/10.1007/978-1-4615-4529-3_15

[27] J. W. Taylor, "Triple seasonal methods for short-term electricity demand forecasting," *European Journal of Operational Research*, vol. 204, no. 1, pp. 139–152, 2010.

[28] K. Kumar, "Time series analysis by state space methods," *Journal Of The Royal Statistical Society Series A-Statistics In Society*, vol. 167, pp. 187–188, 2004.

[29] 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.

[30] Z. Zhao, C. Wang, M. Nokleby, and C. J. Miller, "Improving short-term electricity price forecasting using day-ahead lmp with arima models," in *2017 IEEE Power Energy Society General Meeting*, July 2017, pp. 1–5.

[31] Z. Tan, J. Zhang, J. Wang, and J. Xu, "Day-ahead electricity price forecasting using wavelet transform combined with arima and garch models," *Applied Energy*, vol. 87, no. 11, pp. 3606–3610, 2010.

[32] R. Weron and A. Misiorek, "Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models," *International Journal of Forecasting*, vol. 24, no. 4, pp. 744–763, 2008.

[33] S. J. Koopman, M. Ooms, and M. A. Carnero, "Periodic seasonal reg-arfima-garch models for daily electricity spot prices," *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 16–27, 2007. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1198/016214506000001022

[34] J. Crespo Cuaresma, J. Hlouskova, S. Kossmeier, and M. Obersteiner, "Forecasting electricity spot-prices using linear univariate time-series models," *Applied Energy*, vol. 77, no. 1, pp. 87–106, 2004.

[35] O. Fosso, A. Gjelsvik, A. Haugstad, B. Mo, and I. Wangensteen, "Generation scheduling in a deregulated system. the norwegian case," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 75–81, 1999.

[36] J. Contreras, R. Espinola, F. Nogales, and A. Conejo, "Arima models to predict next-day electricity prices," *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1014–1020, 2003.

[37] J. P. González, A. M. S. Roque, and E. A. Pérez, "Forecasting functional time series with a new hilbertian armax model: Application to electricity price forecasting," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 545–556, Jan 2018.

[38] R. Garcia, J. Contreras, M. van Akkeren, and J. Garcia, "A garch forecasting model to predict day-ahead electricity prices," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 867–874, 2005.

[39] L. Wu and M. Shahidehpour, "A hybrid model for day-ahead price forecasting," *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1519–1530, 2010.

[40] F. Ziel and R. Weron, "Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks," *Energy Economics*, vol. 70, p. 396–420, Feb 2018.

[41] Cartea and M. G. Figueroa, "Pricing in electricity markets: A mean reverting jump diffusion model with seasonality," *Applied Mathematical Finance*, vol. 12, no. 4, pp. 313–335, 2005. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/13504860500117503

[42] R. Weron, "Market price of risk implied by asian-style electricity options and futures," *Energy Economics*, vol. 30, no. 3, pp. 1098–1115, 2008.

[43] S. Trueck, R. Weron, and R. Wolff, "Outlier treatment and robust approaches for modeling electricity spot prices," University Library of Munich, Germany, MPRA Paper, 2007. [Online]. Available: https://EconPapers.repec.org/RePEc:pra:mprapa:4711

[44] J. Janczura and R. Weron, "Modeling electricity spot prices: Regime switching models with price-capped spike distributions," *IDEAS Working Paper Series from RePEc*, 2010. [Online]. Available: http://search.proquest.com/docview/1699082682/

[45] J. Janczura, S. Trück, R. Weron, and R. C. Wolff, "Identifying spikes and seasonal components in electricity spot price data: A guide to robust modeling," *Energy Economics*, vol. 38, no. C, pp. 96–110, 2013.

[46] R. Weron, *Modeling and forecasting electricity loads and prices : a statistical approach*, ser. Wiley finance series. Chichester, England ; Hoboken, NJ: John Wiley Sons, 2006.

[47] R. Kleber, "The nature of statistical learning theory," *The American Mathematical Monthly*, vol. 103, no. 7, p. 616, 1996. [Online]. Available: http://search.proquest.com/docview/203839230/

[48] X. Geng and L. Xie, "Learning the lmp-load coupling from data: A support vector machine based approach," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1127–1138, 2017.

[49] B. Szkuta, L. Sanabria, and T. Dillon, "Electricity price short-term forecasting using artificial neural networks," *IEEE Transactions on Power Systems*, vol. 14, no. 3, 1999. [Online]. Available: http://search.proquest.com/docview/899159901/

[50] S. Anbazhagan and N. Kumarappan, "Day-ahead deregulated electricity market price forecasting using recurrent neural network," *IEEE Systems Journal*, vol. 7, no. 4, pp. 866–872, 2013.

[51] N. Amjady, "Day-ahead price forecasting of electricity markets by a new fuzzy neural network," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 887–896, 2006.

[52] L. Zhang, P. Luh, and K. Kasiviswanathan, "Energy clearing price prediction and confidence interval estimation with cascaded neural networks," *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 99–105, 2003.

[53] N. Amjady and A. Daraeepour, "Design of input vector for day-ahead price forecasting of electricity markets," *Expert Systems With Applications*, vol. 36, no. 10, pp. 12 281–12 294, 2009.

[54] H. Jacobsson, "Rule extraction from recurrent neural networks: Ataxonomy and review," *Neural Computation*, vol. 17, no. 6, pp. 1223–1263, 2005.

[55] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[56] J. Lago, F. De Ridder, and B. De Schutter, "Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms," *Applied Energy*, vol. 221, pp. 386–405, 2018.

[57] S. Zhou, L. Zhou, M. Mao, H.-M. Tai, and Y. Wan, "An optimized heterogeneous structure lstm network for electricity price forecasting," *IEEE Access*, vol. 7, pp. 108 161–108 173, 2019.

[58] C. Chatfield, "What is the 'best' method of forecasting?" *Journal of Applied Statistics*, vol. 15, no. 1, pp. 19–38, 1988. [Online]. Available: https://doi.org/10.1080/02664768800000003

[59] A. Mirakyan, M. Meyer-Renschhausen, and A. Koch, "Composite forecasting approach, application for next-day electricity price forecasting," *Energy Economics*, vol. 66, pp. 228–237, 2017.

[60] O. Valenzuela, I. Rojas, F. Rojas, H. Pomares, L. Herrera, A. Guillen, L. Marquez, and M. Pasadas, "Hybridization of intelligent techniques and arima models for time series prediction," *Fuzzy Sets and Systems*, vol. 159, no. 7, pp. 821–845, 2008.

[61] J. Che and J. Wang, "Short-term electricity prices forecasting based on support vector regression and auto-regressive integrated moving average modeling," *Energy Conversion and Management*, vol. 51, no. 10, pp. 1911–1917, 2010.

[62] L. Xiao, W. Shao, M. Yu, J. Ma, and C. Jin, "Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting," *Applied Energy*, vol. 198, pp. 203–222, 2017.

[63] O. Abedinia, N. Amjady, and H. Zareipour, “A new feature selection technique for load and price forecast of electrical power systems,” *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 62–74, 2017.

[64] K. Zheng, Y. Wang, K. Liu, and Q. Chen, “Locational marginal price forecasting: A componential and ensemble approach,” *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4555–4564, 2020.

[65] O. Abedinia, N. Amjady, M. Shafie-khah, and J. Catalão, “Electricity price forecast using combinatorial neural network trained by a new stochastic search method,” *Energy Conversion and Management*, vol. 105, pp. 642–654, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890415007712

[66] Y. Ma, P. Luh, K. Kasiviswanathan, and E. Ni, “A neural network-based method for forecasting zonal locational marginal prices,” in *IEEE Power Engineering Society General Meeting, 2004.*, 2004, pp. 296–302 Vol.1.

[67] D. T. Sandwell, “Biharmonic spline interpolation of geos-3 and seasat altimeter data,” *Geophysical Research Letters*, vol. 14, no. 2, pp. 139–142, 1987.

[68] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” 2016.

[69] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction beyond mean square error,” 2015.

[70] E. Denton, S. Chintala, A. Szlam, and R. Fergus, “Deep generative image models using a laplacian pyramid of adversarial networks,” 2015.

[71] X. Zhou, Z. Pan, G. Hu, S. Tang, and C. Zhao, “Stock market prediction on high-frequency data using generative adversarial nets,” *Mathematical Problems in Engineering*, vol. 2018, pp. 1–11, 04 2018.

[72] M. Claesen and B. D. Moor, “Hyperparameter search in machine learning,” 2015.

[73] M. Geurts, G. E. P. Box, and G. M. Jenkins, “Time series analysis: Forecasting and control,” *Journal of Marketing Research*, vol. 14, no. 2, p. 269, 1977.

[74] W. Polasek, “Time series analysis and its applications: With r examples, third edition by robert h. shumway, david s. stoffer,” *International Statistical Review*, vol. 81, no. 2, pp. 323–325, 2013.

[75] MATLAB, “Estimate parameters of armax model,” 2019. [Online]. Available: https://www.mathworks.com/help/ident/ref/armax.html

[76] “Tensorflow,” 2019. [Online]. Available: https://www.tensorflow.org/

[77] L. Wang, Z. Zhang, and J. Chen, “Short-term electricity price forecasting with stacked denoising autoencoders,” *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2673–2681, 2017.

[78] Jau-Jia Guo and P. B. Luh, "Improving market clearing price prediction by using a committee machine of neural networks," *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1867–1876, Nov 2004.

[79] A. Motamedi, H. Zareipour, and W. D. Rosehart, "Electricity price and demand forecasting in smart grids," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 664–674, June 2012.

[80] P. Mandal, T. Senjyu, K. Uezato, and T. Funabashi, "Several-hours-ahead electricity price and load forecasting using neural networks," in *IEEE Power Engineering Society General Meeting, 2005*, June 2005, pp. 2146–2153 Vol. 3.

[81] A. M. Gonzalez, A. M. S. Roque, and J. Garcia-Gonzalez, "Modeling and forecasting electricity prices with input/output hidden markov models," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 13–24, Feb 2005.

[82] X. Geng and L. Xie, "A data-driven approach to identifying system pattern regions in market operations," in *2015 IEEE Power Energy Society General Meeting*, July 2015, pp. 1–5.

[83] Y. Ji, R. J. Thomas, and L. Tong, "Probabilistic forecasting of real-time lmp and network congestion," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 831–841, March 2017.

[84] K. Hubicka, G. Marcjasz, and R. Weron, "A note on averaging day-ahead electricity price forecasts across calibration windows," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 1, pp. 321–323, 2019.

[85] F. Ziel, "Forecasting electricity spot prices using lasso: On capturing the autoregressive intraday structure," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4977–4987, 2016.

[86] Y. Zhu, R. Dai, G. Liu, Z. Wang, and S. Lu, "Power market price forecasting via deep learning," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 4935–4939.

[87] "Short-term forecast for locational marginal pricing (lmp) data sets," in *2018 North American Power Symposium (NAPS)*, 2018, pp. 1–5.

[88] S. K. Aggarwal, L. M. Saini, and A. Kumar, "Electricity price forecasting in deregulated markets: A review and evaluation," *International Journal of Electrical Power & Energy Systems*, vol. 31, pp. 13 – 22, 2009.

[89] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," 2015.

[90] PJM Interconnection, "Operating agreement of pjm interconnection, l.l.c." 2011. [Online]. Available: pjm.com/directory/merged-tariffs/oa.pdf

[91] J. F. Toubeau, T. Morstyn, J. Bottieau, K. Zheng, D. Apostolopoulou, Z. De Grève, Y. Wang, and F. Vallée, "Capturing spatio-temporal dependencies in the probabilistic forecasting of distribution locational marginal prices," *IEEE Transactions on Smart Grid*, pp. 1–1, 2020.

[92] Z. Zhang and M. Wu, "Real-time locational marginal price forecasting using generative adversarial network," in *2020 IEEE SmartGridComm Conference*, 2020.

[93] ——, "Locational marginal price forecasting using convolutional long-short term memory-based generative adversarial network," in *2021 IEEE Power Energy Society General Meeting (PESGM)*, 2021, pp. 1–5.

[94] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[95] J. Liang and W. Tang, "Ultra-short-term spatiotemporal forecasting of renewable resources: An attention temporal convolutional network-based approach," *IEEE Transactions on Smart Grid*, vol. 13, no. 5, pp. 3798–3812, 2022.

[96] M. F. Azam and M. S. Younis, "Multi-horizon electricity load and price forecasting using an interpretable multi-head self-attention and eemd-based framework," *IEEE Access*, vol. 9, pp. 85 918–85 932, 2021.

[97] M. Massaoudi, H. Abu-Rub, S. S. Refaat, A. Ali Al-Kuwari, and T. Huang, "Locational marginal electricity price forecasting-based self-attention mechanism and simulated annealing optimizer using big data," in *2021 10th International Conference on Renewable Energy Research and Application (ICRERA)*, 2021, pp. 391–396.

[98] Z. Zhang and M. Wu, "Energy price prediction considering generation bids variation: A two-stage convolutional long short-term memory approach," in *2022 IEEE Power Energy Society General Meeting (PESGM)*, 2022, pp. 1–5.

[99] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15 084–15 097, 2021.

[100] Z. Zhang and M. Wu, "Predicting real-time locational marginal prices: A gan-based approach," *IEEE Transactions on Power Systems*, vol. 37, no. 2, pp. 1286–1296, 2022.

[101] T. Orfanogianni and G. Gross, "A general formulation for lmp evaluation," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1163–1173, 2007.

[102] X. Geng and L. Xie, "Learning the lmp-load coupling from data: A support vector machine based approach," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1127–1138, March 2017.

[103] C. Tofallis, “A better measure of relative prediction accuracy for model selection and model estimation,” *Journal of the Operational Research Society*, vol. 66, no. 8, pp. 1352–1362, 2015.

[104] R. Ferrero, S. Shahidehpour, and V. Ramesh, “Transaction analysis in deregulated power systems using game theory,” *IEEE Transactions on Power Systems*, vol. 12, no. 3, pp. 1340–1347, 1997.

[105] “Google colabratory,” 2019. [Online]. Available: https://colab.research.google.com/notebooks/welcome.ipynb/