HISTORICAL ANALYSIS AND ESTIMATION TOOL FOR ENERGY TRADING

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Sharath Chandra Sambaraju

In Partial Fulfillment
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

December 2013

Fargo, North Dakota

North Dakota State University
Graduate School

Title

HISTORICAL ANALYSIS AND ESTIMATION TOOL FOR ENERGY TRADING

By

SHARATH CHANDRA SAMBARAJU

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Simone Ludwig
Chair

Dr. Rui Dai

Dr. Luis Del Rio

Approved:

04/14/2014
Date

Dr. Brian Slator
Department Chair

ABSTRACT

Many forms of commodities are traded in today's financial world, the most commonly used commodity, 'Electricity', is no exception. Electricity is mostly traded on a day to day basis, where the buyers and suppliers agree on a fixed price for electricity to be generated and dispatched. These allow the markets to adjust themselves to efficient pricing based on the simple supply and demand principle.

The objective of this paper is to understand the day ahead market data and collecting historical market data and storing in to a RDBMS database, and design and program a windows-based application using C# programming language, to query the RDBMS in transact SQL to display the market data in a format that is easy to read. Use of this tool would allow deeper analysis and provide the ability to estimate possible profitability for future trading by calculating the variances, standard deviation, upper bounds, lower bounds, averages.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my advisers, Dr. Simone Ludwig and Dr. Kendall Nygard, for their continued support throughout this paper. I am grateful for the ideas and suggestions given by Dr. Nygard, and the amount of guidance he gave is enormous. Also, special thanks go to my advisory committee members.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

This chapter focuses on the problem description, energy markets, understanding the market data and the New York Independent System Operator (NYISO) [1] energy market we chose to build the trading tool for.

1.1. Problem Description

The energy markets allow traders and suppliers to buy and sell electricity based on the demand and supply principles. The energy market operates o; complex sets of data which include the power supply zones which are geographically mapped, the day-ahead prices of electricity that suppliers are ready to generate power measured in dollars per megawatt hours (MWh); and the real time prices that change due to issues that arise in power production and supply [6] [8].

A tool that arranges the data in an easily readable format and allows traders to query and perform mathematical calculations to identify trends quickly could be invaluable to traders and would increase their ability to make well informed decisions. In this paper we are going to collect and store historical data in to an RDBMS and provide a graphical user interface (GUI) for the traders to use.

1.2. Energy Markets

Commodities such as Oil, Gas, Electricity and other sources of energy fall under the Energy Markets [7]. Where electricity or power is traded is referred to as the electricity market [6] [8].

1.2.1. Electricity Market

Electricity is one of the commodities that is needed all the time, but once generated electricity has to be readily consumed as storing it can be quiet expensive. Energy markets exist to make electricity cheaper for the users as well as to provide more electricity where the generators

simply follow the principle ‘economies of scale’, that is to generate more electricity at cheaper costs and still be profitable. Energy generators have to take in to account, all the risks that would break the flow of electricity like a power plan failure, broken transmission lines, inclement weather and natural calamities and the predicted peak load to generate enough electricity, usually more supply than demand. Energy markets allow the power generation companies to fix prices in to the future and accept bids from independent traders to buy the output at a fixed cost [6] [8].

1.2.2. Trading Markets

There are many electricity markets in the United States like, PJM Interconnection LLC (PJM), Electrical Reliability Council of Texas (ERCOT), New York Independent System Operator (NYISO), Midwest ISO, California ISO and New England ISO [6]. Each of these markets are associated with the power generated in different geographical areas of this country. In this way, PJM Interconnection LLC (PJM) is a Regional Transmission Organization (RTO) which is part of the Eastern Interconnection grid operating an electric transmission system serving all or parts of Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia and the District of Columbia. ERCOT serves all of Texas, NYISO serves all of New York State, Midwest ISO serves all of the Midwestern states and California ISO serves the state of California.

1.2.3. NYISO

NYISO [1] is one of the energy markets in the USA, where power generating companies and traders come together to take part in the trading process of selling and buying electricity. At NYISO, traders can purchase electricity in the day ahead market at different zones which are nothing but geographical locations in the New York state. The following are the tradable zones in

NYISO: West, Genesee, Central, Mohawk Valley, North, Capital, Hudson Valley, Millwood and Dunwood [10] as shown in Figure 1.

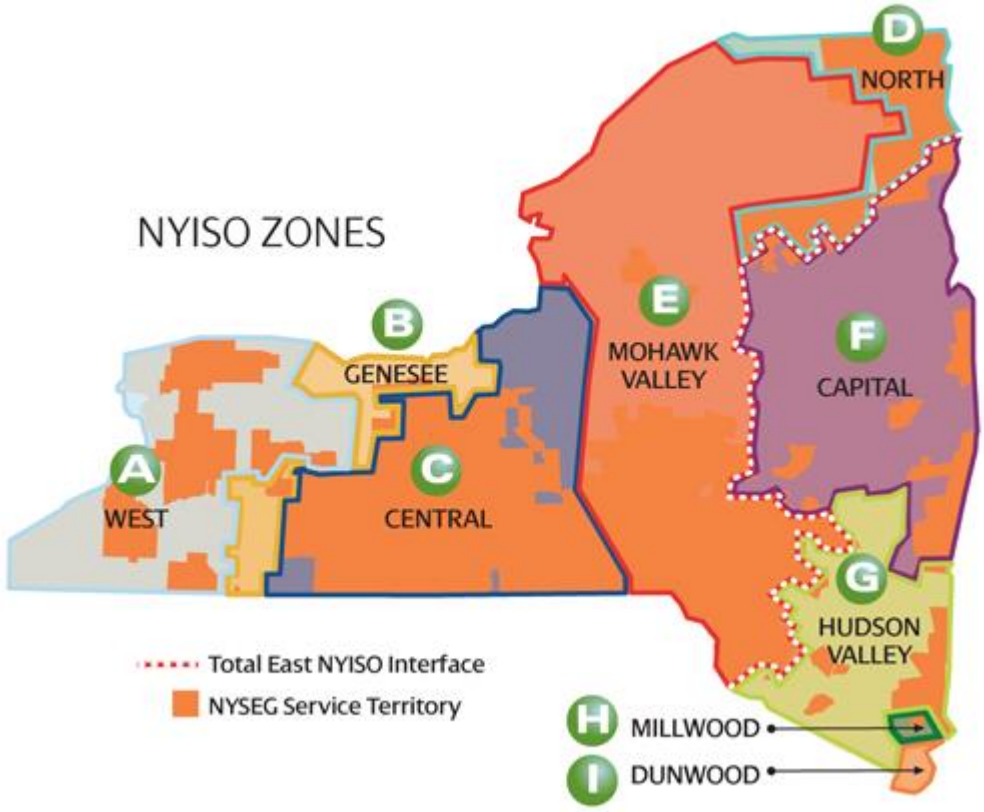


Figure 1. NYISO Map.

1.3. Market Data

NYISO as well as other markets have a repository of historical data like pricing data, load data and constraint information that is available to the public. We are going to use the day ahead and the real time pricing data for different zones for displaying and performing mathematical calculations on the data. Market data for NYISO is hosted at the NYISO's website [13].

2. RELATED WORK

Any company that wants to participate in electricity trading would require tools that would help analyzing the pricing data and help traders in making informed decisions to buy electricity or not to generate revenue [10] [18] [19]. Historical analysis and prediction tools are developed and used by many trading companies today. Analysis tools are researched and developed by all trading companies and the logic and uses of the tools are kept secret as the tools are designed to maximize profits and reduce risks. As well as due to the high R & D costs involved in building these tools make them publicly inaccessible. But again the logical setup or implementation principles of any of these tools are highly guarded from public domains and remain unknown due to the proprietary trademarks of the solution providers.

While these programs are highly guarded, many private companies, like 'Sungard Financial Systems', 'OATI' 'CME Group', 'VENTYX', 'TIBCO' etc. develop and sell similar tools in the open market. These tools are used by traders to run simulations, building graphs, and conducting risk analysis.

Here are some of the related works and tools provided by private companies that can be used for energy trading.

SUNGARD: Sungard Financial Systems [21] offers a tool for traders called as 'ALIGNE'. This product has various features providing the ability of:

- Risk management and risk analysis through pricing and analysis models.
- Asset valuation and optimization and stress testing.
- Measurement of various market risks such as Value at Risk (VaR), Earnings at Risk (EaR), Cash Flow at Risk (CFaR), and What-If VaR Analysis Tools.

- Business Intelligence: Provides a set of business intelligence and reporting tools that can be used with any common data model and databases including advanced cubes. It also provides the traders with dashboards that can be used to show real-time alerts and data feeds to deliver consolidated views, graphical representations of the data, forecasts, real-time market data and projected performance metrics.

OATI: OATI provides various sets of tools for trading electricity. Here is a brief description of a few tools and their capabilities.

- OATI WebTrader [16] tool provides a comprehensive solution for financial and physical trading, scheduling, risk, and settlements in energy markets. As well as position reports, robust reporting features, and data calculation features.
- OATI WebIntelligence [14] tool gives energy traders options to gather data, organize data, display and create custom tabular or graphic reports, and analyze data by built-in advanced forecasting service to predict LMPs based on historical data or forecasted loads. Using meter and weather data the tool can automatically export the forecasted value to any external system. As well as give the ability to download the data into their personal databases or upload specific data into the tool for analysis.
- OATI WebRisk [15] tool allows traders and trading companies to manage and measure various risk types like market risk, credit risk, and enterprise-wide risk and the ability to simulate "what-if" scenarios to optimize trading strategies.

CME Group: CME Group provides a 'Quikstrike Option Valuation Tool' [3] that is available across all asset classes and features data on current and historical volatility by strike, pricing analysis data, spread analysis and risk graphics, delta sheets etc.

VENTYX: Ventyx provides the ‘Energy Market Intelligence, Forecasts, & Analytics’ [24] tool that delivers hourly forecasts of mid- to long-term electric market prices easily, determine the effects of transmission congestion, fuel costs, generator availability, bidding behavior, and load growth on market prices. This tool contains the historic energy markets data that provides traders analysis tools that benchmark, conduct planning and market analysis.

TIBCO: TIBCO provides the ‘SPOTFIRE’ [22] analytics tool that can be used to analyze large amounts of data in a smaller time frame by giving the ability to interactively query by providing a GUI, visualize, aggregate, filter, and drill into large datasets. And lets the traders use data from various sources for analysis and decision making, It also provides risk analysis for better understanding of the risk profile and better management and control of risk.

3. APPROACH

In this chapter we are going to create an executable plan to find a solution for implementing and building a historical analysis tool. The approach is shown in Figure 2 as a step by step process.

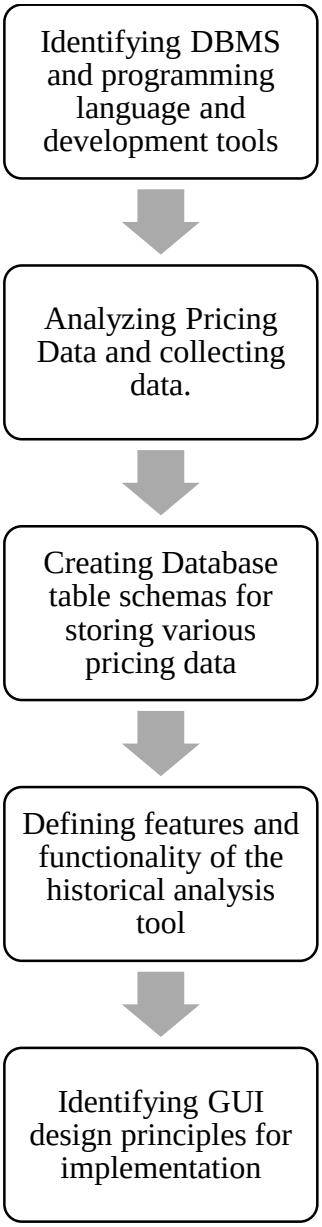


Figure 2. Approach Flow Chart.

We need to identify the right tools for solving the problem. Understanding the pricing data and designing a DBMS is the first step towards solving this problem. The importance of accurate historical data is paramount and a good database design which follows the ACID (Atomicity, Consistency, Isolation, and Durability) properties helps in maintaining stability and consistency of the application.

Once the data is cleaned and stored into the DBMS, the other steps would involve the GUI design that follows the globally accepted user interface design principles such as Aesthetically pleasing, Clarity, Compatibility, Comprehensibility, Configurability, Consistency, Control, Efficiency and Simplicity. The above user interface design principles are taken into consideration to make the GUI easy to user, presentable and efficient to handle the user inputs and display the data with accuracy ensuring readability.

The third step would involve formulating mathematical functions that would be performed on the data to give users the ability to aggregate historical data into views. Here we are also going to design the acceptable inputs for the tool combined with the mathematical function that would be added to the conditional statements in Transact SQL for querying data. Thus, providing the users with a complete and usable GUI that enables creation of various scenarios on the same data.

3.1. Development Tools

The tools more commonly used for the design and implementation of databases is, Microsoft SQL Server Management Studio 2012 [11] while the tool used for design, development and coding of GUI is Microsoft Visual Studio 2010 [12]. The programming language used is C# [2].

3.2. Pricing Data

Energy Markets like NYISO provide pricing data for various zones, also called LBMP (Location Based Marginal Pricing) in downloadable Comma Separated Values (CSV) files as well as Portable Document Format (PDF) and Hyper Text Markup Language (HTML) formats [13].

3.2.1. Day Ahead and Real Time Prices

The day-ahead LBMP's contain price (in $/MWh) for combined energy, losses, and transmission congestion determined on an hourly basis and the traders will contract to buy electricity at the day ahead prices only. The other pricing data is the real-time LBMP's which, as the result of the unavoidable demand and supply issues, transmission issues and other constraints such as weather, generator failures, natural calamities, is usually higher or lower than the day-ahead prices traders agreed to pay. The difference between these two sets represents the loss or a gain in monetary value.

The images in Figure 3 and Figure 4 below are the examples of day-ahead and real time data respectively for a zone in New York generated by NYISO [13].

Day Ahead Market Zonal LBMP

--- LBMP $ --- Marginal Cost of Losses --- Marginal Cost of Congestion

Zonal Prices

Name PTID	00:00 EST	01:00 EST	02:00 EST	03:00 EST	04:00 EST	05:00 EST	06:00 EST	07:00 EST	08:00 EST	09:00 EST	10:00 EST	11:00 EST
CAPITL 61757	34.81	33.04	24.98	23.90	32.04	38.99	49.08	51.15	47.11	49.96	50.47	48.92
	1.58	1.58	1.68	1.69	1.65	2.18	3.45	2.98	3.08	3.00	2.54	2.23
	-11.30	-10.05	-1.12	0.00	-8.96	-8.82	-1.40	-3.75	0.00	-4.13	-10.63	-11.85
CENTRL 61754	23.82	23.10	22.80	22.74	23.05	29.74	45.51	46.88	46.82	45.99	41.27	38.65
	0.48	0.43	0.49	0.53	0.49	0.64	1.11	1.02	0.66	0.60	0.52	0.49
	-1.42	-1.26	-0.14	0.00	-1.12	-1.11	-0.18	-1.43	-2.13	-2.55	-3.43	-3.32
DUNWOD 61760	33.01	31.42	25.35	24.50	30.61	37.78	50.04	52.33	49.26	51.11	50.01	48.29
	2.37	2.27	2.31	2.29	2.27	3.00	4.73	5.02	5.24	5.10	4.51	4.32
	-8.71	-7.74	-0.86	0.00	-6.90	-6.80	-1.08	-2.89	0.00	-3.18	-8.19	-9.13

Name PTID	12:00 EST	13:00 EST	14:00 EST	15:00 EST	16:00 EST	17:00 EST	18:00 EST	19:00 EST	20:00 EST	21:00 EST	22:00 EST	23:00 EST
CAPITL 61757	48.38	46.76	47.25	36.00	45.90	62.28	56.33	47.43	42.70	46.22	38.86	37.96
	2.14	2.01	1.90	2.15	2.81	3.81	3.49	2.87	2.50	2.46	2.21	1.52
	-13.35	-12.87	-15.23	-1.27	-0.47	-0.77	-0.03	-0.44	-0.59	-6.48	-4.58	-14.13
CENTRL 61754	36.09	34.87	33.03	33.88	44.52	60.52	55.68	46.44	41.26	38.90	33.11	24.42
	0.52	0.51	0.42	0.49	0.64	1.09	1.06	0.88	0.75	0.52	0.35	0.34
	-2.67	-2.48	-2.49	-0.80	-1.26	-1.73	-1.80	-1.44	-0.90	-1.09	-0.68	-1.77
DUNWOD 61760	47.26	45.82	45.62	46.34	54.07	74.72	59.73	55.34	52.30	46.90	39.35	35.50
	4.08	4.02	3.76	4.21	5.33	6.98	6.49	5.47	4.99	4.62	3.75	2.30
	10.29	-9.92	-11.74	-9.54	-6.13	-10.05	-0.42	-5.75	-7.69	-4.99	-3.53	-10.89

Figure 3. Day Ahead Prices Example.

Time Weighted/Integrated Real Time LBMP - Zonal

LBMP $ --- Marginal Cost of Losses --- Marginal Cost of Congestion

Zonal Prices

Name PTID	00:00 EST	01:00 EST	02:00 EST	03:00 EST	04:00 EST	05:00 EST	06:00 EST	07:00 EST	08:00 EST	09:00 EST	10:00 EST	11:00 EST
CAPITL 61757	21.41 0.10 -19.85	25.46 0.56 -17.25	14.22 0.68 -4.05	27.50 1.67 -2.11	23.04 1.53 0.00	25.23 1.69 -0.26	73.81 4.78 -2.46	38.09 2.54 0.00	39.83 2.38 -4.29	34.89 2.14 -1.54	34.85 1.84 -6.29	40.65 2.05 -8.95
CENTRL 61754	3.98 0.03 -2.49	9.93 0.12 -2.17	10.15 0.16 -0.51	24.39 0.41 -0.27	21.92 0.42 0.00	23.72 0.42 -0.03	73.60 1.16 -5.86	56.37 0.66 -20.17	39.84 0.74 -5.94	33.41 0.58 -1.62	30.17 0.49 -2.96	34.75 0.46 -4.64
DUNWOD 61760	16.89 0.16 -15.28	21.80 0.88 -13.28	13.66 1.07 -3.12	27.99 2.65 -1.63	23.99 2.48 0.00	26.15 2.68 -0.20	75.58 7.11 -1.90	39.39 3.85 0.00	40.14 3.68 -3.30	35.82 3.43 -1.18	34.63 3.06 -4.84	40.01 3.48 -6.89

Name PTID	12:00 EST	13:00 EST	14:00 EST	15:00 EST	16:00 EST	17:00 EST	18:00 EST	19:00 EST	20:00 EST	21:00 EST	22:00 EST	23:00 EST
CAPITL 61757	38.25 2.11 -5.45	34.62 1.84 -5.17	23.59 1.39 -1.34	19.52 1.21 -0.26	31.38 1.78 -2.63	36.10 1.96 -4.77	29.70 1.69 -1.85	31.98 1.86 -1.72	33.94 2.14 -0.72	33.45 1.81 -5.19	21.09 0.60 -11.86	2.58 0.17 0.00
CENTRL 61754	35.80 0.46 -4.65	33.34 0.44 -5.28	21.44 0.24 -0.33	18.31 0.23 -0.02	29.21 0.42 -1.83	29.96 0.45 -0.14	26.55 0.39 0.00	29.35 0.51 -0.44	41.14 0.56 -9.49	29.30 0.33 -2.52	10.29 0.09 -1.58	2.44 0.02 0.00
DUNWOD 61760	38.53 3.65 -4.19	34.91 3.31 -3.98	24.42 2.52 -1.03	23.61 2.21 -3.35	64.13 3.27 -33.89	93.46 3.41 -60.69	52.00 3.04 -22.80	52.60 3.34 -20.87	43.59 3.71 -8.81	34.93 3.12 -5.35	18.73 0.98 -9.13	2.69 0.28 0.00

Figure 4. Real Time Prices Example.

In Figure 3 and Figure 4, you can see the prices are arranged by zone with a name and a unique ID also referred to as PTID and the prices for different hours in the day starting from 00:00 to 23:00. Hour 00:00 refers to the whole hour starting from 12 AM to 1 AM and henceforth ending with hour 23:00 covering the entire hour ending at 11:59 PM. To make things easier we will represent the hours as Hour Ending's (HE) that would translate the hours from 0 to 23 I to HE01 to HE24 while storing the data in the database.

3.2.2. Causes for Real Time Price Fluctuations

Apart from demand and supply issues, transmission issues and other constraints such as weather, generator failures and natural calamities can affect prices of electricity. Other reasons that also cause real time price fluctuations are the different days of the week like weekdays and weekends where the energy loads drastically change when people and business, factories and other industries change their power consumption needs.

Larger businesses might still continue to operate all days of the week but might close on specific holidays changing the demand for electricity. And the other major factor is peak and non-peak hours of the day, where the energy needs of businesses as well as people change quiet rapidly as the activity during the peak hours is much larger than the non-peak hours. The peak hours are considered as the time between 8 AM to 11 PM and non-peak from 11 PM to 7:59 AM.

3.2.3. Profits and Loss

As we understand that the traders buy for the day ahead or the next day and are eventually tied to selling the electricity at real time prices, profit and loss depends on what the buying price was and the difference in the real time selling price. So, if the real time prices are higher than the day ahead prices, there is profit made in terms of ($/MWh) for the number of units of MW's bought or vice versa.

3.3. Database Design

The primary purpose of database design is to make sure that the data is arranged in tabular format containing rows and columns, while maintaining the integrity of the data by avoiding redundancies, incorrect or wrong data and normalizing the data to maintain its consistency and integrity [5].

Purpose of the database – The purpose of the database is to store the pricing data that would be used in the historical analysis and estimation tool. The tool should be able to populate the day ahead and real time prices and give the user the ability to select the combination zones as well as the hours and specific days of the week to recreate various scenarios for example, analyzing data for a particular hour of a particular day over a period of a few years.

Find and organize the information required – The data we must have for the data base is the dates, real time, day ahead prices, information about the zones such as the names of the zones and unique ID. Figure 5 shows the database schema.

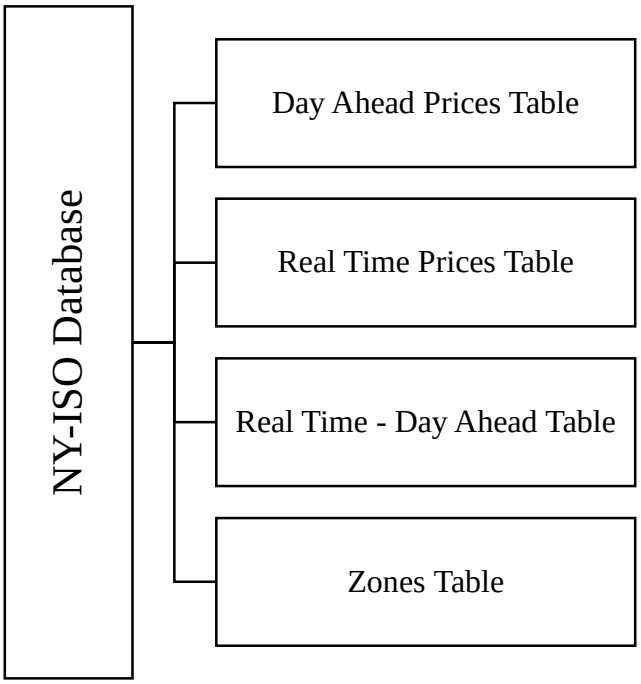


Figure 5. Database Structure.

Divide the information into tables – From the data available four tables can be created one each for the Day Ahead data, Real Time data, Zonal data and an additional table which would store the difference of prices between the Real Time and Day ahead prices for each and every day, zone

by the hour in order to reduce the response time when the user just wants to look at the difference or the average difference of prices for a given date range for a set of zones and hours.

Define Columns – The columns identify various types of information pertaining to the table schema. The real time, day ahead and the difference table would have a similar schema with the same number of columns due to the identical sets of data they hold. The list of columns include [NID], [Day], [HE01], [HE02], [HE03], [HE04], [HE05], [HE06], [HE07],[HE08], [HE09], [HE10], [HE11], [HE12], [HE13], [HE14], [HE15], [HE16], [HE17], [HE18], [HE19], [HE20], [HE21], [HE22], HE23], [HE24], [Changed], [PriceType]. Where NID denoted the Node ID, Changed denoted the timestamp when ever the data is modified for maintenance purposes and Pricec Type denotes if the prices are for Real Time or Day Ahead or the Difference (RT-DA). A separate table for the Zones should also be designed which contains the following columns [NID] denotes the node ID an alias for each Zone, [Name] denotes the zones name, [NYID] denoted the unique PTID and [Changed] denoted the timestamp for maintenance.

Primary Keys - The primary key is a column, or a set of columns, that is used to uniquely identify each row. As we have a large dataset, in order to avoid redundancies while inserting, deleting and reading data we need to have primary keys for each and every table in the database. The primary key for the day ahead, real time and the difference tables would be the combination of the Date and Node ID (NID) as shown in the images below. And the Node Name table would have distinct node names or zones hence the Name column is designated as the primary key as shown in Figure 6.

Column Name	Data Type	Allow Nulls
NID	int	☐
Name	varchar(30)	☐
NYID	nchar(15)	☐
Changed	datetime	☐
		☐

Figure 6. Node Name Table Schema.

Figure 7 below shows the schema for various pricing types created in the NYISO database.

Column Name	Data Type	Allow Nulls
NID	int	☐
Day	date	☐
HE01	decimal(18, 2)	☑
HE02	decimal(18, 2)	☑
HE03	decimal(18, 2)	☑
HE04	decimal(18, 2)	☑
HE05	decimal(18, 2)	☑
HE06	decimal(18, 2)	☑
HE07	decimal(18, 2)	☑
HE08	decimal(18, 2)	☑
HE09	decimal(18, 2)	☑
HE10	decimal(18, 2)	☑
HE11	decimal(18, 2)	☑
HE12	decimal(18, 2)	☑
HE13	decimal(18, 2)	☑
HE14	decimal(18, 2)	☑
HE15	decimal(18, 2)	☑
HE16	decimal(18, 2)	☑
HE17	decimal(18, 2)	☑
HE18	decimal(18, 2)	☑
HE19	decimal(18, 2)	☑
HE20	decimal(18, 2)	☑
HE21	decimal(18, 2)	☑
HE22	decimal(18, 2)	☑
HE23	decimal(18, 2)	☑
HE24	decimal(18, 2)	☑
Changed	datetime	☐
PriceType	varchar(5)	☑

Figure 7. Day Ahead, Real Time, Difference Table's Schema.

3.4. Functionality and Risk Prediction

We need to define the functionality and the requirements of the tool before we start designing the Graphical User Interface. Here we will take into account the different ways the data can be modified and viewed by applying various mathematical functions and also define an approach for predicting risks that would help in estimating profits and losses for day ahead trading.

3.4.1. Functionality

For a trader who is interested in analyzing historical data there is no easy alternative as it would take enormous amounts of time to go through pages and pages of pricing information. For a user to search for a particular set of data is much more time consuming and cumbersome. So, the most important functionality of the tool would be to have a variety of inputs that would qualify for the search and query criteria such as the ability to select specific zones of interest, combined with specific days and even particular hours of the days over a large date range.

Once the selection criteria is established, the other important aspect is to condense the huge amounts of data into meaningful mathematically applicable formats such as averages of day ahead, real time, differences and combinations of all three types of prices when the traders can look at the average day ahead prices, real time and the differences. Giving them an idea about the average profit and loss over a particular scenario.

3.4.2. Display Criteria

We will discuss the different criteria's that can be emulated into the tool so that the traders can avoid using time taking process of loading data into Excel sheets and manipulating the data. For the ease of reading lets represent the Day Ahead as 'DA', Real Time as 'RT' and the Difference of Real Time and Day Ahead (RT-DA) as 'DIFF'. So, we can give the traders the options to view just the Real Time data or the Day ahead data or the Difference data with the selection criteria.

And to view the Day ahead Averages, Real Time Averages, and Difference Averages. This can be achieved by taking the inputs from the tool and building a SQL query to display the data into a data grid.

3.4.3. Predicting Profit and Loss

For predicting profit and loss we are going to use the following values: averages of the Difference between Real Time prices and Day Ahead prices (RT-DA), calculate Standard Deviation of the RT-DA prices and calculate the 95% confidence interval of the prices and get the upper bound and the lower bounds of the prices over a large data set.

Averages: Average means the sum of the listed entries divided by the size of the total entries. Averages provide a distinct view of the pricing data in our tool, as the traders can easily spot the higher RT-DA prices on an average in making a decision. So, we shall consider the use of RT_DA average for the prediction.

Variance and Standard Deviation: Variance measures the spread of a set of numbers and the square root of variance is called the standard deviation. A small variance indicates that the data points tend to be very close to the mean or the average value, while a high variance indicates that the data points are very spread out from the mean and from each other.

The variance is typically designated as Var(X),$\sigma 2_x$ or simply σ^2 (pronounced "sigma squared") [23]. And can also be defined as the average of the squared differences from the mean. The expression for the variance can be expanded:

$$\text{Var(X)} = E[X^2 - 2XE[X] + (E[X])^2] \quad (1)$$

$$= E[X^2 - 2E[X]E[X] + (E[X])^2]$$

$$= E[X^2] - (E[X])^2]$$

A mnemonic for the above expression is "mean of square minus square of mean". And Standard Deviation [20] is designed by the symbol σ (the Greek letter sigma) and can be expanded as

Standard deviation = SQRT (Variance) or

$$\sigma = \sqrt{\text{Var}(X)} \quad (2)$$

Standard Deviation along with Confidence Intervals [4] are highly used in the financial sectors to indicate the reliability of an estimate. A confidence interval gives an estimated range of values which is likely to include an unknown population parameter (Prediction Price Difference), the estimated range being calculated from a given set of sample data.

The selection of a confidence level for an interval determines the probability that the confidence interval produced will contain the true parameter value. In our probability model we will choose a 95% confidence interval. A 95% confidence interval covers 95% of the normal curve the probability of observing a value outside of this area is less than 0.05.

We can calculate the Lower and Upper bounds of the 95% confidence interval by using the formulae below.

$$\text{Lower endpoint} = \bar{X} - 1.96\ \frac{\sigma}{\sqrt{n}} \quad (3)$$

$$\text{Upper endpoint} = \bar{X} + 1.96\ \frac{\sigma}{\sqrt{n}}$$

Where $\bar{X}$ represents the sample mean and n is the sample size.

Three Day Average: As the tool is capable of creating various scenarios, we wanted to consider the three day RT-DT average (the three occurrences before the Day Ahead) into the estimation formula, as this immediate data might be missing from the statistical analysis. And the possibility of a trend in prices that closely follow the previous days values.

3.4.4. Estimation Formula

In order to predict a Real Time price a day before its occurrence we need to assign, different probability values to the mathematical functions above as per their prediction abilities and categorizing the final probability into risks.

The estimation pseudo code for our tool is as given below.

Assume probability P = 0;

IF Average (RealTime-DayAhead) > 0;

P = P + 0.3;

IF (UpperBound – LowerBound) > 0;

P = P + 0.5;

IF (Threeday RealTime-DayAhead Average) > 0;

P = P + 0.2;

IF P > 0.8

Definite Buy;

IF 0.6<P<0.8

Buy;

IF 0.45<P<0.6

Moderate Risk;

IF 0.3<P<0.45

Risky;

IF P<0.3

High Risk;

We are going to apply this logic in the C# code to display the risk of buying electricity for a Day Ahead price based on various simulations by the user.

3.5. Graphical User Interface Design

For the GUI design we will take into consideration the various industry standards, such as aesthetically pleasing, clarity, compatibility, comprehensibility, configurability, consistency, control, directness and efficiency.

3.5.1. Design Principles

The following are the Interface design principles and standards [9] we are going to use in designing the analysis tool:

- Aesthetically Pleasing: Creating meaningful elements and grouping them together while maintaining a simple design makes a GUI easy to understand and use.
- Clarity: The GUI should have clarity in its elements, functions words and text.
- Compatibility: The GUI should be compatible with the user's views, about the tasks and functions of the interface.
- Comprehensibility: The interface should be easy to learn and the user should easily know of what to do, what to look at, how to do and where on the interface and the order of actions should be easily recollect able.
- Consistency: The system should look and feel the same across the board. Each and every element should have a similar look, have similar uses, operate similarly while every function should always yield the same results and the positions of the elements should be static.

- Control: The user must always control the interaction on the interface. And every action should be performed quickly under explicit user requests and should be error free. The means to achieve goals should be flexible and compatible with the user's skills, experiences, habits and preferences.
- Efficiency: The interface should have various features to make it efficient to use such as, minimal eye and hand movements, easy transactions and simpler flows between various systems, shorter navigation paths and anticipated users wants and needs whenever possible.

4. IMPLEMENTATION

For the implementation of the Historical Analysis Tool structured query language (SQL) was used to query the database tables and return the outputs. C# was selected as the programming language to build the GUI and the logic for the tool. SQL is widely used as a standard querying mechanism by a majority of software developers and C# is an object oriented programming language also widely used in conjunction with Microsoft Visual Studio for various types of software development across the world. These selection have a huge impact on the project as they are very efficient and reliable and also save time in developing software. They also are easier to manage and edit in the future for further enhancements.

We followed a standard three-tier architecture, where the three layers are the GUI, Data Access Layer (logic or code) or the Business Logic and the Data Layer (Database).

4.1. Graphical User Interface (GUI)

Following the GUI design principles as mention in the Approach and the requirements, let us implement the design. Figure 8 shows the content and features of the tool that was built in the implementation phase.

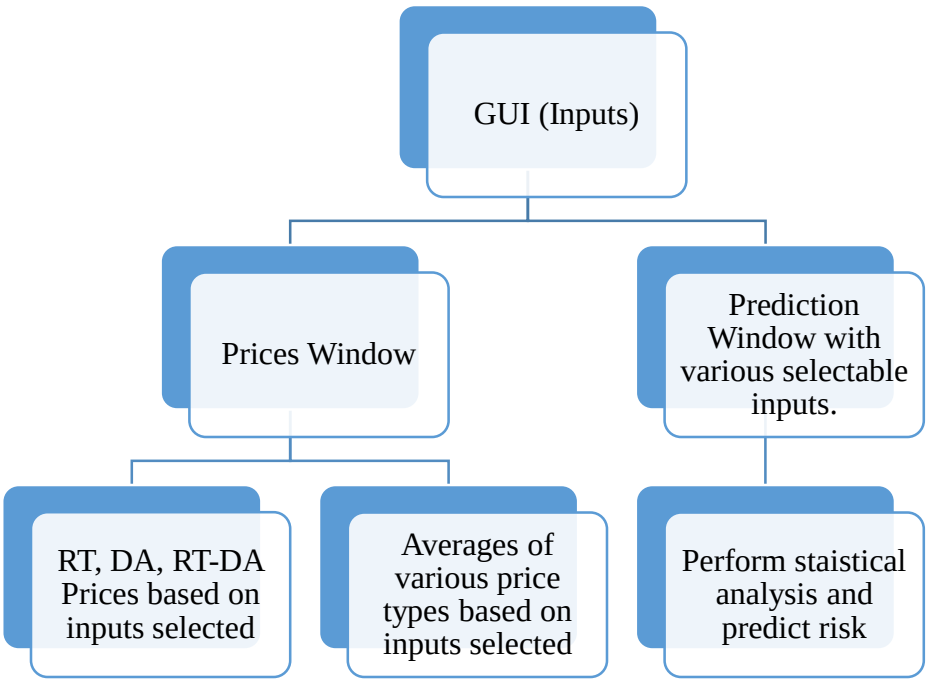


Figure 8. Functionality of the Tool.

4.1.1. Selection Criteria or Inputs

A menu of inputs to be used in simulations or data analysis are presented below for the various function based on the requirements of having the ability to choose any one or various Zones in NYISO, Having a very detailed Date and Time options where the users can choose a start date and an end date, different days of the week, and different hours and particular months. And a prices sub category for the users to view different prices or a combination of them at once.

Here Figure 9 shows the windows that present Zone Selection choices; Figure 10 shows Prices Sub Categories; and Figure 11 shows the date time criteria.

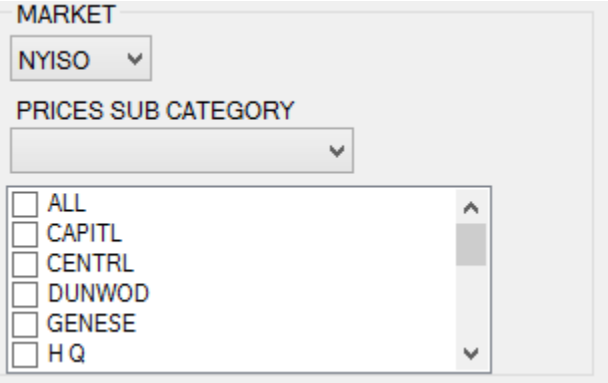


Figure 9. Zone Selection Criteria.

Here the Zones are displayed as a collection of CheckedListBoxes using C#. The user can select any combination of zones or deselect them while simulating a scenario. The select Zones will be inputs for the SQL query.

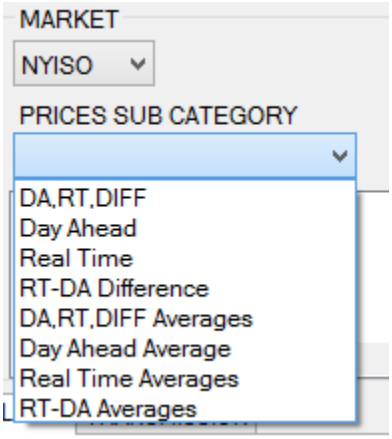


Figure 10. Price Sub Category Selection Criteria.

Figure 10 shows the prices sub category where the user has the ability to choose the different price types and Averages of those price types they wish to see. 'DA,RT,DIFF' and 'DA,RT,DIFF Averages' gives the ability to display all three price types arranged by the Zone and Day in a tabular format. And the other selections are self-explanatory.

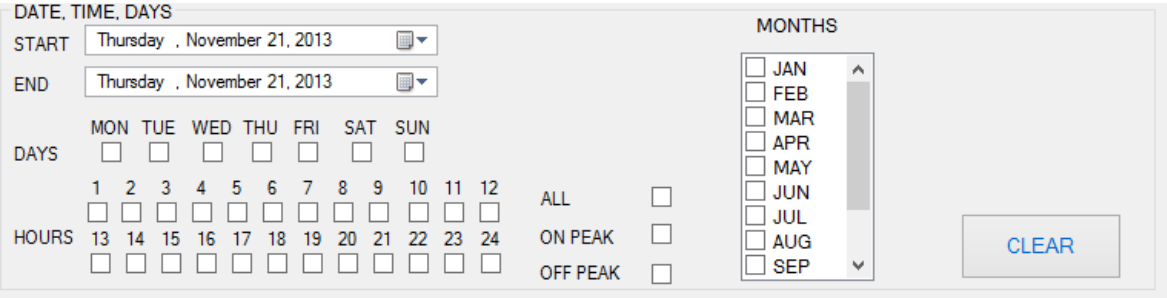


Figure 11. Date, Time and Hours Selection Criteria.

Figure 11 shows the date, time, day's selector where the user is given various options to choose the month, date, time and day of the week to simulate various criteria's. Having the option to choose Months is important as the traders can look for data pertaining to various seasons, as the electricity usage varies from season to season. As an example, people use more electricity on very cold or very hot days.

The 'ALL' checkbox works as a select all function, that selects all the hours and all the months and all the days in the selection criteria. The 'ON PEAK' checkbox when checked will select the hours from 8 to 23 and the 'OFF PEAK' when checked selects the hours 1 to 7 and 24. This adds more functionality where the users can just select one checkbox instead of clicking on all related hours.

The 'CLEAR' button works a clear all function where all the users' previous selections will be deselected for a fresh new set of inputs.

All of this logic is implemented in C# in the data access layer, where the button checks and unchecks are configured to trigger their respective functions like selecting all the checkboxes or clearing them.

4.1.2. LMP and Prediction Buttons

With the selection criteria in place, we need to define two functions for the user to analyze data and they are setup as buttons that can be clicked to obtain the results.

LMP: The LMP function takes the various inputs selected by the user and returns the pricing data by displaying it in a Data Grid View (tabular format). For every Price Sub Category selected a different SQL query is executed to return the right output data. So, the data access layer or the logic is coded in C# [2] to check for the price selections and dynamically generate an executable SQL query that will be executed on the database with the other inputs as conditions.

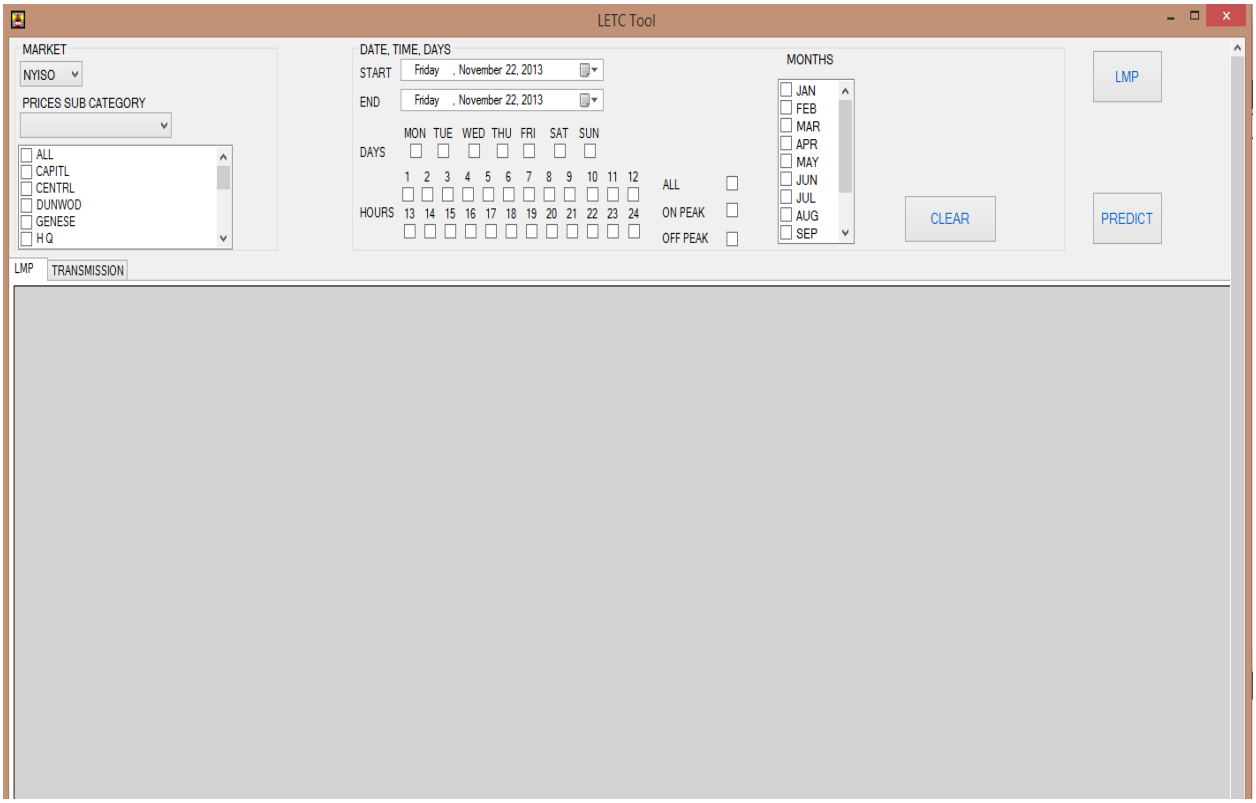
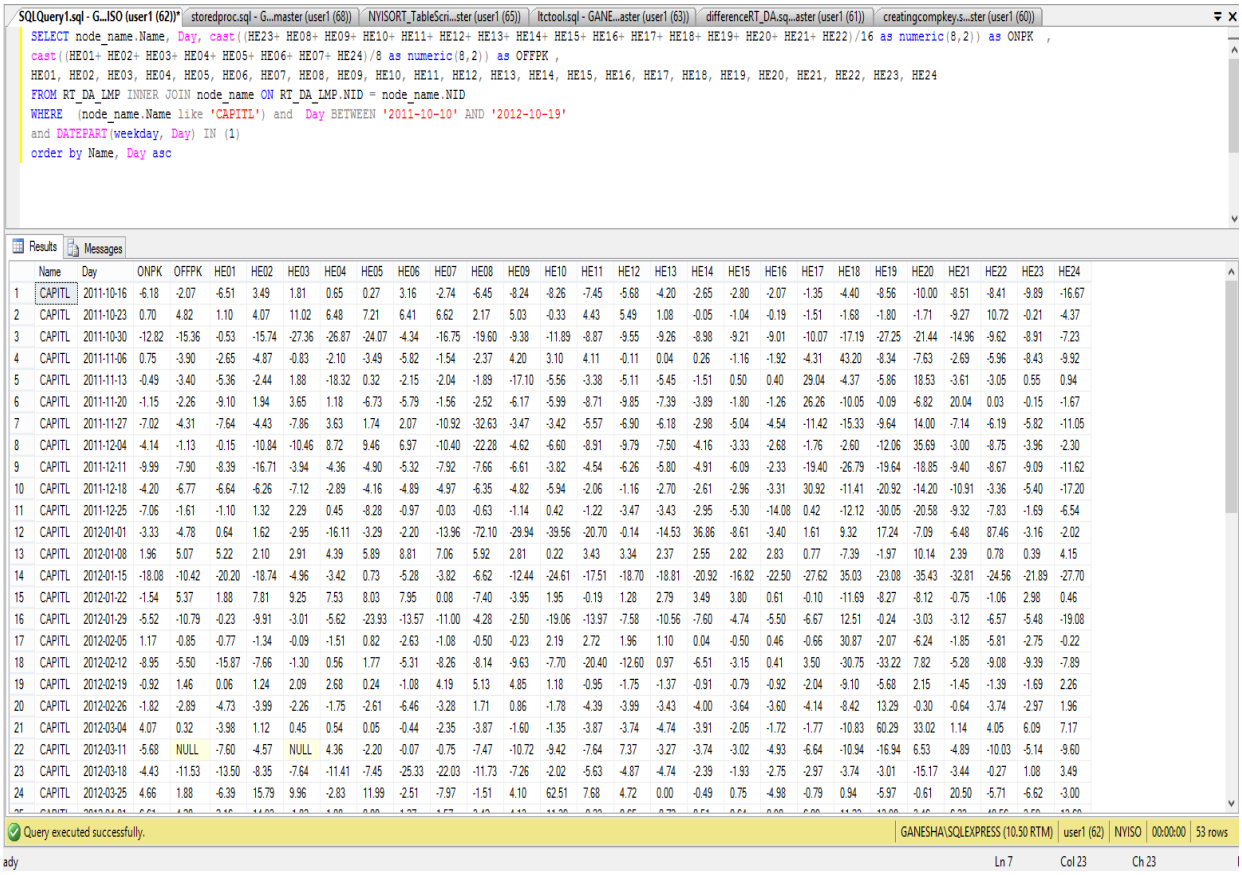


Figure 12. GUI of Historical Analysis Tool.

Example of a SQL query based on the inputs to display the ON Peak Average, Off Peak average, prices for all the hours from the Difference table for Zone CAPTIL for the date range of all the day between '2011-10-10' AND '2012-10-19' and Sundays only is shown in Figure 13.



```
SELECT node_name.Name, Day, cast((HE23+ HE08+ HE09+ HE10+ HE11+ HE12+ HE13+ HE14+ HE15+ HE16+ HE17+ HE18+ HE19+ HE20+ HE21+ HE22)/16 as numeric(8,2)) as ONPK  ,
cast((HE01+ HE02+ HE03+ HE04+ HE05+ HE06+ HE07+ HE24)/8 as numeric(8,2)) as OFFPK ,
HE01, HE02, HE03, HE04, HE05, HE06, HE07, HE08, HE09, HE10, HE11, HE12, HE13, HE14, HE15, HE16, HE17, HE18, HE19, HE20, HE21, HE22, HE23, HE24
FROM RT_DA_LMP INNER JOIN node_name ON RT_DA_LMP.NID = node_name.NID
WHERE  (node_name.Name like 'CAPITL') and  Day BETWEEN '2011-10-10' AND '2012-10-19'
and DATEPART(weekday, Day) IN (1)
order by Name, Day asc
```

	Name	Day	ONPK	OFFPK	HE01	HE02	HE03	HE04	HE05	HE06	HE07	HE08	HE09	HE10	HE11	HE12	HE13	HE14	HE15	HE16	HE17	HE18	HE19	HE20	HE21	HE22	HE23	HE24
1	CAPITL	2011-10-16	-6.18	-2.07	-6.51	3.49	1.81	0.65	0.27	3.16	-2.74	-6.45	-8.24	-8.26	-7.45	-5.68	-4.20	-2.65	-2.80	-2.07	-1.35	-4.40	-8.56	-10.00	-8.51	-8.41	-9.89	-16.67
2	CAPITL	2011-10-23	0.70	4.82	1.10	4.07	11.02	6.48	7.21	6.41	6.62	2.17	5.03	-0.33	4.43	5.49	1.08	-0.05	-1.04	-0.19	-1.51	-1.68	-1.80	-1.71	-9.27	10.72	-0.21	-4.37
3	CAPITL	2011-10-30	-12.82	-15.36	-0.53	-15.74	-27.36	-26.87	-24.07	-4.34	-16.75	-19.60	-9.38	-11.89	-8.87	-9.55	-9.26	-8.98	-9.21	-9.01	-10.07	-17.19	-27.25	-21.44	-14.96	-9.62	-8.91	-7.23
4	CAPITL	2011-11-06	0.75	-3.90	-2.65	-4.87	-0.83	-2.10	-3.49	-5.82	-1.54	-2.37	4.20	3.10	4.11	-0.11	0.04	0.26	-1.16	-1.92	-4.31	43.20	-8.34	-7.63	-2.69	-5.96	-8.43	-9.92
5	CAPITL	2011-11-13	-0.49	-3.40	-5.36	-2.44	1.88	-18.32	0.32	-2.15	-2.04	-1.89	-17.10	-5.56	-3.38	-5.11	-5.45	-1.51	0.50	0.40	29.04	-4.37	-5.86	18.53	-3.61	-3.05	0.55	0.94
6	CAPITL	2011-11-20	-1.15	-2.26	-9.10	1.94	3.65	1.18	-6.73	-5.79	-1.56	-2.52	-6.17	-5.99	-8.71	-9.85	-7.39	-3.89	-1.80	-1.26	26.26	-10.05	-0.09	-6.82	20.04	0.03	-0.15	-1.67
7	CAPITL	2011-11-27	-7.02	-4.31	-7.64	-4.43	-7.86	3.63	1.74	2.07	-10.92	-32.63	-3.47	-3.42	-5.57	-6.90	-6.18	-2.98	-5.04	-4.54	-11.42	-15.33	-9.64	14.00	-7.14	-6.19	-5.82	-11.05
8	CAPITL	2011-12-04	-4.14	-1.13	-0.15	-10.84	-10.46	8.72	9.46	6.97	-10.40	-22.28	-4.62	-6.60	-8.91	-9.79	-7.50	-4.16	-3.33	-2.68	-1.76	-2.60	-12.06	35.69	-3.00	-8.75	-3.96	-2.30
9	CAPITL	2011-12-11	-9.99	-7.90	-8.39	-16.71	-3.94	-4.36	-4.90	-5.32	-7.92	-7.66	-6.61	-3.82	-4.54	-6.26	-5.80	-4.91	-6.09	-2.33	-19.40	-26.79	-19.64	-18.85	-9.40	-8.67	-9.09	-11.62
10	CAPITL	2011-12-18	-4.20	-6.77	-6.64	-6.26	-7.12	-2.89	-4.16	-4.89	-4.97	-6.35	-4.82	-5.94	-2.06	-1.16	-2.70	-2.61	-2.96	-3.31	30.92	-11.41	-20.92	-14.20	-10.91	-3.36	-5.40	-17.20
11	CAPITL	2011-12-25	-7.06	-1.61	-1.10	1.32	2.29	0.45	-8.28	-0.97	-0.03	-0.63	-1.14	0.42	-1.22	-3.47	-3.43	-2.95	-5.30	-14.08	0.42	-12.12	-30.05	-20.58	-9.32	-7.83	-1.69	-6.54
12	CAPITL	2012-01-01	-3.33	-4.78	0.64	1.62	-2.95	-16.11	-3.29	-2.20	-13.96	-72.10	-29.94	-39.56	-20.70	-0.14	-14.53	36.86	-8.61	-3.40	1.61	9.32	17.24	-7.09	-6.48	87.46	-3.16	-2.02
13	CAPITL	2012-01-08	1.96	5.07	5.22	2.10	2.91	4.39	5.89	8.81	7.06	5.92	2.81	0.22	3.43	3.34	2.37	2.55	2.82	2.83	0.77	-7.39	-1.97	10.14	2.39	0.78	0.39	4.15
14	CAPITL	2012-01-15	-18.08	-10.42	-20.20	-18.74	-4.96	-3.42	0.73	-5.28	-3.82	-6.62	-12.44	-24.61	-17.51	-18.70	-18.81	-20.92	-16.82	-22.50	-27.62	35.03	-23.08	-35.43	-32.81	-24.56	-21.89	-27.70
15	CAPITL	2012-01-22	-1.54	5.37	1.88	7.81	9.25	7.53	8.03	7.95	0.08	-7.40	-3.95	1.95	-0.19	1.28	2.79	3.49	3.80	0.61	-0.10	-11.69	-8.27	-8.12	-0.75	-1.06	2.98	0.46
16	CAPITL	2012-01-29	-5.52	-10.79	-0.23	-9.91	-3.01	-5.62	-23.93	-13.57	-11.00	-4.28	-2.50	-19.06	-13.97	-7.58	-10.56	-7.60	-4.74	-5.50	-6.67	12.51	-0.24	-3.03	-3.12	-6.57	-5.48	-19.08
17	CAPITL	2012-02-05	1.17	-0.85	-0.77	-1.34	-0.09	-1.51	0.82	-2.63	-1.08	-0.50	-0.23	2.19	2.72	1.96	1.10	0.04	-0.50	0.46	-0.66	30.87	-2.07	-6.24	-1.85	-5.81	-2.75	-0.22
18	CAPITL	2012-02-12	-8.95	-5.50	-15.87	-7.66	-1.30	0.56	1.77	-5.31	-8.26	-8.14	-9.63	-7.70	-20.40	-12.60	0.97	-6.51	-3.15	0.41	3.50	-30.75	-33.22	7.82	-5.28	-9.08	-9.39	-7.89
19	CAPITL	2012-02-19	-0.92	1.46	0.06	1.24	2.09	2.68	0.24	-1.08	4.19	5.13	4.85	1.18	-0.95	-1.75	-1.37	-0.91	-0.79	-0.92	-2.04	-9.10	-5.68	2.15	-1.45	-1.39	-1.69	2.26
20	CAPITL	2012-02-26	-1.82	-2.89	-4.73	-3.99	-2.26	-1.75	-2.61	-6.46	-3.28	1.71	0.86	-1.78	-4.39	-3.99	-3.43	-4.00	-3.64	-3.60	-4.14	-8.42	13.29	-0.30	-0.64	-3.74	-2.97	1.96
21	CAPITL	2012-03-04	4.07	0.32	-3.98	1.12	0.45	0.54	0.05	-0.44	-2.35	-3.87	-1.60	-1.35	-3.87	-3.74	-4.74	-3.91	-2.05	-1.72	-1.77	-10.83	60.29	33.02	1.14	4.05	6.09	7.17
22	CAPITL	2012-03-11	-5.68	NULL	-7.60	-4.57	NULL	4.36	-2.20	-0.07	-0.75	-7.47	-10.72	-9.42	-7.64	7.37	-3.27	-3.74	-3.02	-4.93	-6.64	-10.94	-16.94	6.53	-4.89	-10.03	-5.14	-9.60
23	CAPITL	2012-03-18	-4.43	-11.53	-13.50	-8.35	-7.64	-11.41	-7.45	-25.33	-22.03	-11.73	-7.26	-2.02	-5.63	-4.87	-4.74	-2.39	-1.93	-2.75	-2.97	-3.74	-3.01	-15.17	-3.44	-0.27	1.08	3.49
24	CAPITL	2012-03-25	4.66	1.88	-6.39	15.79	9.96	-2.83	11.99	-2.51	-7.97	-1.51	4.10	62.51	7.68	4.72	0.00	-0.49	0.75	-4.98	-0.79	0.94	-5.97	-0.61	20.50	-5.71	-6.62	-3.00

Figure 13. SQL Query Example 1.

Example of a SQL query to display average prices of Hour Ending 8, 9, 12, 15, 18, 21 and 23 for the Zone CAPTIL over a data range of all the weekdays between '2018-04-10' AND '2013-06-10' as shown in Figure 14.

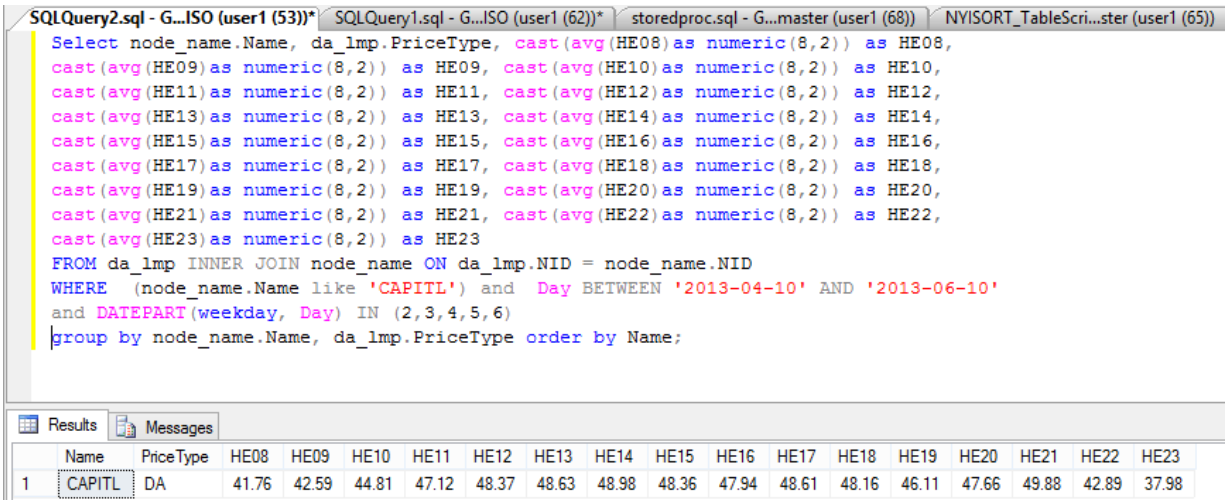


Figure 14. SQL Query Example 2.

PREDICT: The Predict button is defined to open a new Predict widow, where the user gets an additional selection criteria that is completely independent from the main LMP window. This additional GUI is designed to make the tool more user friendly and to improve the overall efficiency. So, the users can view the LMP prices in the first window and click on the PREDICT button to open a new window to perform statistical analysis and get a RISK value of buying electricity at Day Ahead prices.

The predict button creates a new window on every click. Giving the user the ability to multitask working with two windows as well as save time by creating different scenarios in multiple windows. For example the user can view the 'DA, RT, DIFF' for a period of time in the first window and use the Prediction window to run the statistical analysis. So, the user can look at the risks and view the whole set of data on the first window side by side. Figure 15 is the screen shot of the Prediction windows GUI.

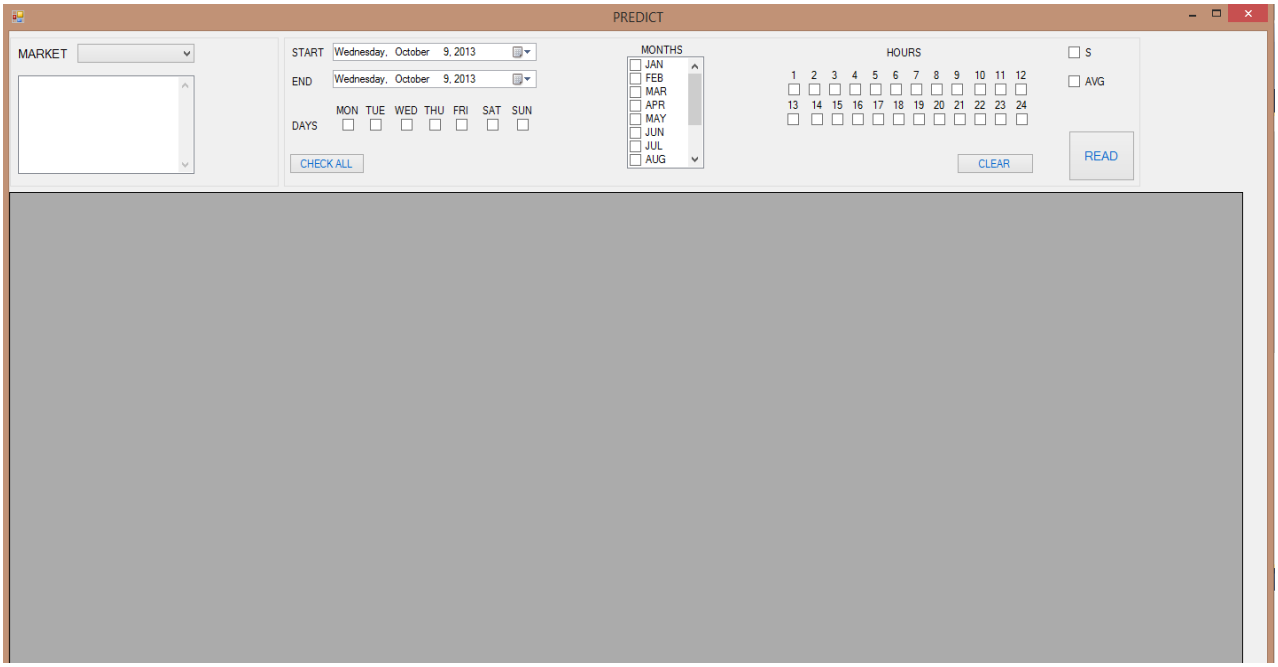


Figure 15. GUI of Prediction Window.

4.2. *Data Access Layer or Business Logic*

The logical part of the project is implemented in C# [2] an objected oriented programming language [17]. The business logic is the part of the program that encodes the real-world business rules that determine how data can be created, displayed, stored, and changed. It is contrasted with the remainder of the software which might be concerned with lower-level details of managing a database or displaying the user interface, system infrastructure, or generally connecting various parts of the program.

The C# Forms classes are used in generating the GUI containing the various Checkboxes and Buttons. And C# events are used to derive the log behind clicking various function buttons and checkboxes. And based on the selected inputs strings and string builders are used in dynamically creating an executable SQL Query that will be sent to the SQL Server for execution

returning valid outputs. The output is then populated on the GUI using Data Sets and Data Grid Views which store the data in a tabular format. The data in the data grid views can be further processed and identified using the Rows and Column classes provided by the object oriented programming framework.

Example Code for an event that is raised when the ‘OFF PEAK’ box is check in the date and time selector panel. This event calls the offpeakcheck() function that checks the hours from 1 to 7 and 24 while unchecking the rest of them on the GUI as shown in Figure 16.

Figure 16. Coding Example 1.

Example code for dynamically building the inputs for the SQL queries to be executed on the server as stored procedures by dynamically assigning parameters with values is shown in the Figure 17.

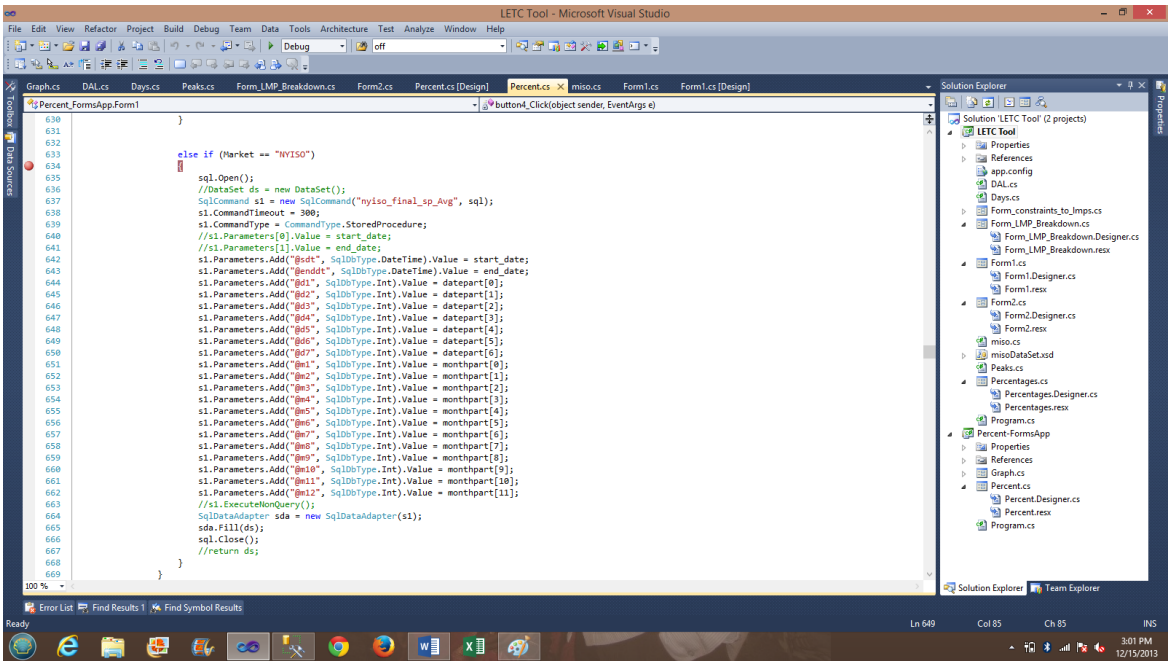


Figure 17. Coding Example 2.

A SQL Query that can be dynamically initiated and stored on the SQL Server. Stored Procedures are highly efficient and reduce the data transferred from the application to the server and reduce redundancies. In the above example, a SQL connection is established and the various inputs are passed on as parameters to the Stored Procedure.

4.3. Data Layer

The data layer represents the implementation of the databases on the servers. As the database schemas have been created earlier, we need to create the table schemas and load the data from the CSV files from NYISO into their respective tables.

Here is an example of the script for creating the Day Ahead price table on the SQL Server as shown in Figure 18.

```
/****** Object:  Table [dbo].[DA_LMP]    Script Date: 7/7/2013 11:03:50 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[RT_LMP](
    [NID] [int] NOT NULL,
    [Day] [date] NOT NULL,
    [HE01] [decimal](18, 4) NULL,
    [HE02] [decimal](18, 4) NULL,
    [HE03] [decimal](18, 4) NULL,
    [HE04] [decimal](18, 4) NULL,
    [HE05] [decimal](18, 4) NULL,
    [HE06] [decimal](18, 4) NULL,
    [HE07] [decimal](18, 4) NULL,
    [HE08] [decimal](18, 4) NULL,
    [HE09] [decimal](18, 4) NULL,
    [HE10] [decimal](18, 4) NULL,
    [HE11] [decimal](18, 4) NULL,
    [HE12] [decimal](18, 4) NULL,
    [HE13] [decimal](18, 4) NULL,
    [HE14] [decimal](18, 4) NULL,
    [HE15] [decimal](18, 4) NULL,
    [HE16] [decimal](18, 4) NULL,
    [HE17] [decimal](18, 4) NULL,
    [HE18] [decimal](18, 4) NULL,
    [HE19] [decimal](18, 4) NULL,
    [HE20] [decimal](18, 4) NULL,
    [HE21] [decimal](18, 4) NULL,
    [HE22] [decimal](18, 4) NULL,
    [HE23] [decimal](18, 4) NULL,
    [HE24] [decimal](18, 4) NULL,
    [Changed] [datetime] NOT NULL,
 CONSTRAINT [pk_newRTconstraint] PRIMARY KEY CLUSTERED
(
    [NID] ASC,
    [Day] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

Figure 18. SQL Script for Creating a Table.

After creating the schemas for the price tables, we need to write an algorithm that would parse the data from the CSV files containing the pricing data. So, as we are developing a tool for historical analysis, we downloaded the CSV files from NYISO for a time period starting from January 2008 to June 2013, totaling 66 months of pricing data.

Here is an example of the parsing algorithm written in C# that would access the CSV files stored in a folder on the hard drive as shown in Figure 19.

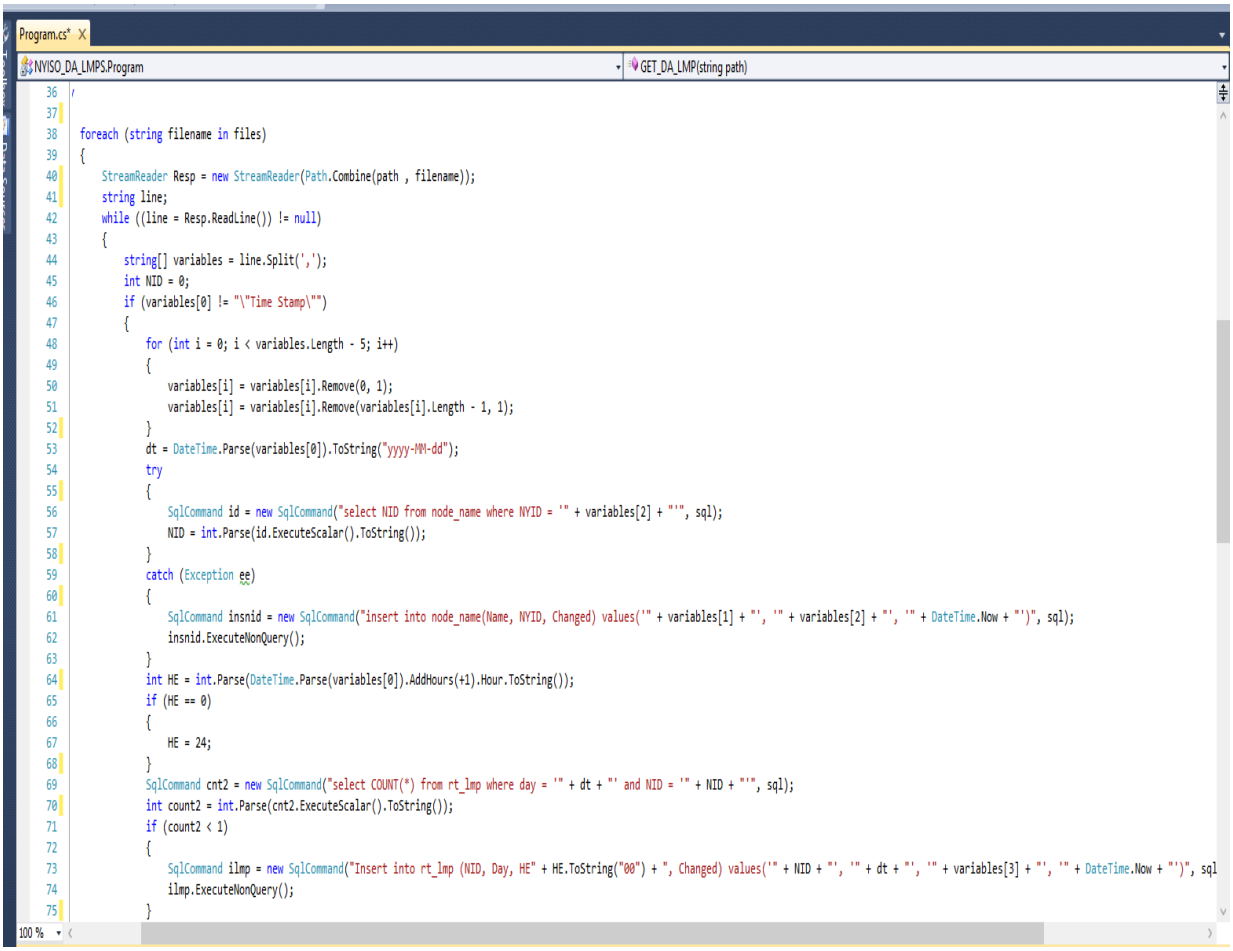


Figure 19. Coding Example 3.

Here Figure 20 is a screenshot of the data base table for the Day Ahead prices after the data is inserted showing a total of 30030 rows of data.

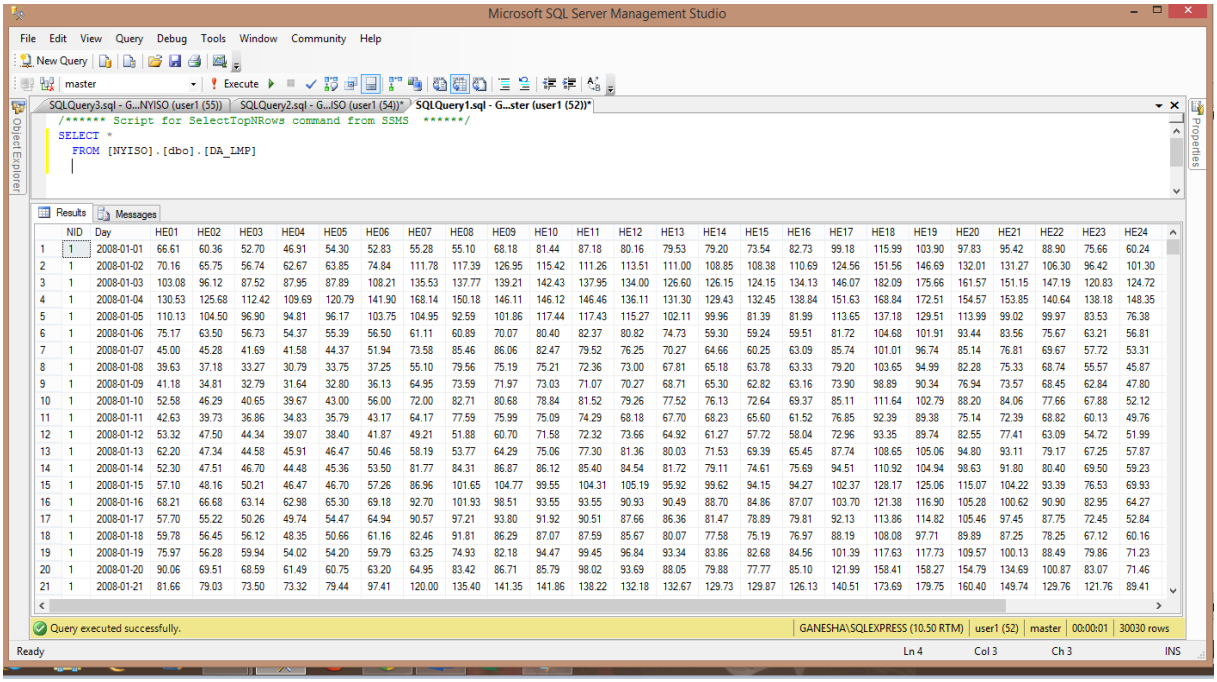


Figure 20. Day Ahead Pricing Data from the Database Table.

Here Figure 21 is a screenshot of the data base table for the Real Time prices after the data is inserted showing a total of 30015 rows of data.

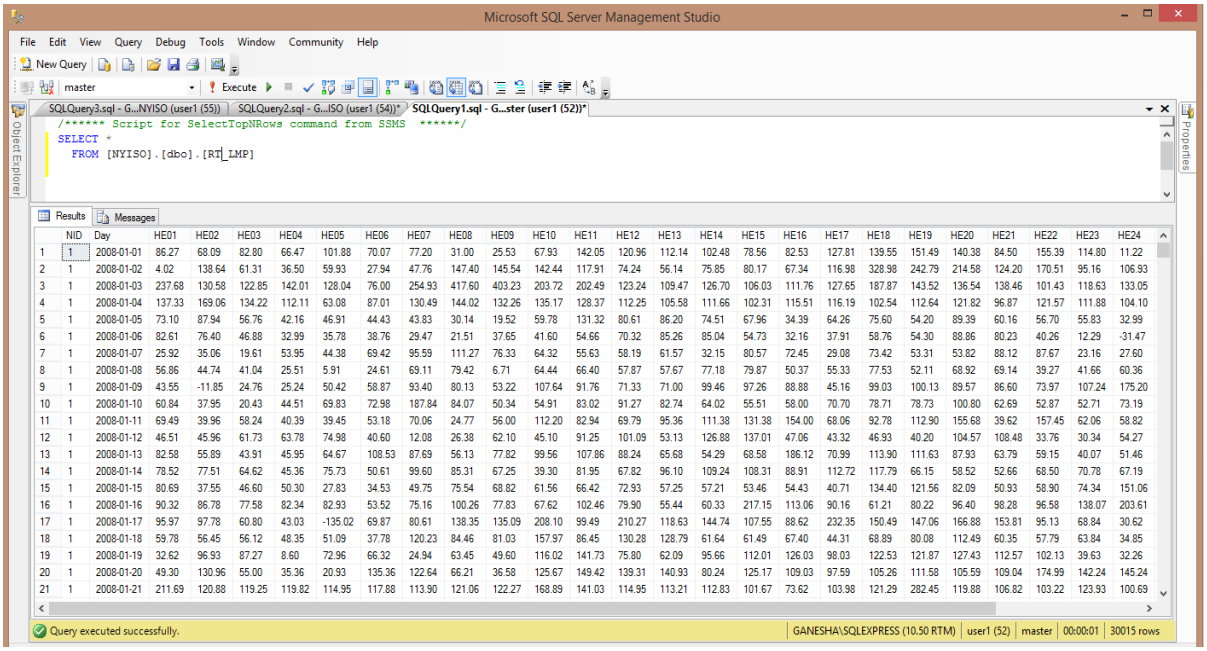
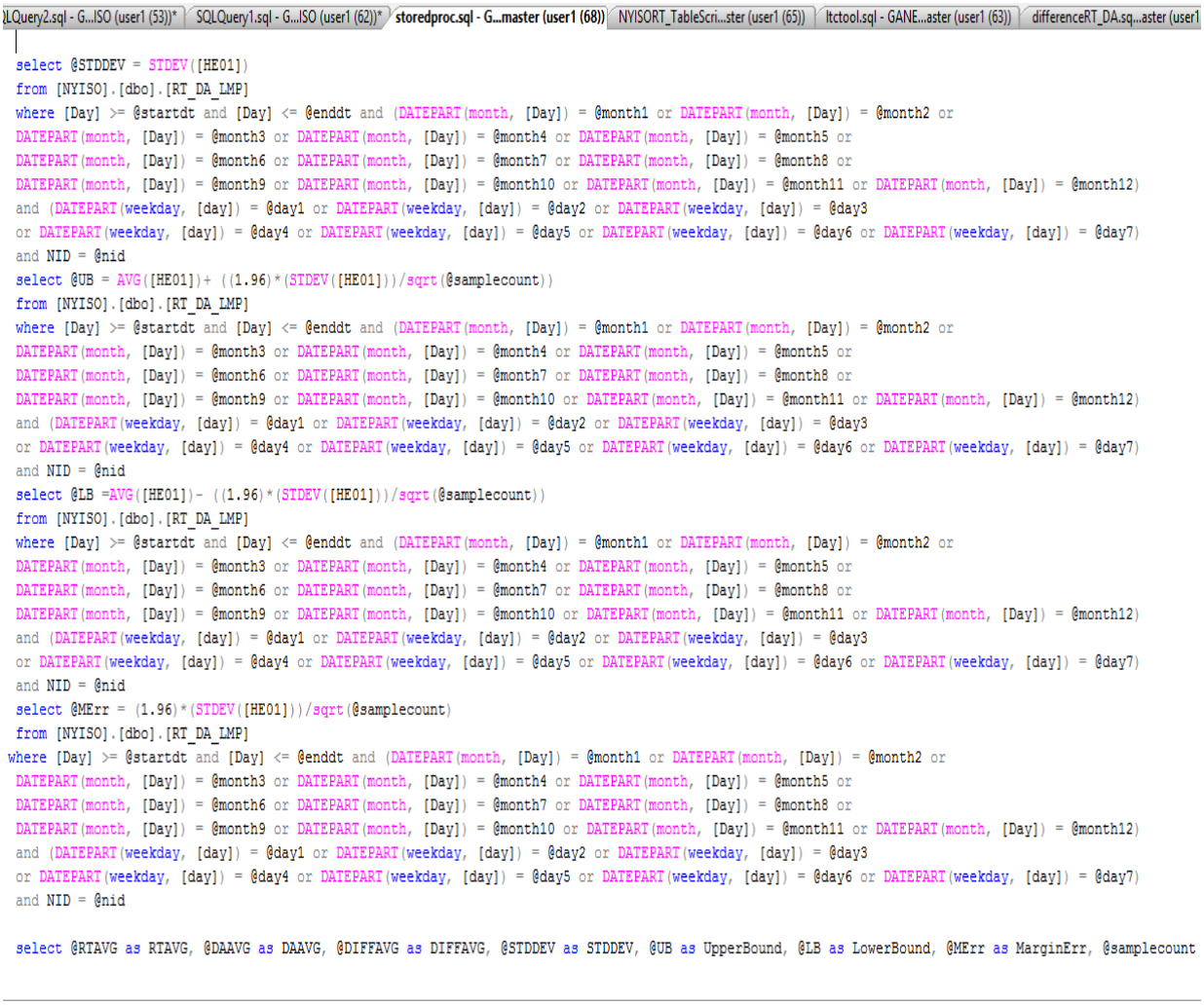


Figure 21. Real Time Pricing Data from the Database Table.

Here is an example part of a stored procedure that will be stored on the database and can be accessed by the business access layer dynamically to return the standard deviation and other mathematical function values based on the price data that can be used for calculating the risk factor as shown in Figure 22.



```
)LQuery2.sql - G...ISO (user1 (53))* | SQLQuery1.sql - G...ISO (user1 (62))* | storedproc.sql - G...master (user1 (68)) | NYISORT_TableScri...ster (user1 (65)) | ltctool.sql - GANE...aster (user1 (63)) | differenceRT_DA.sq...aster (user1

select @STDDEV = STDEV([HE01])
from [NYISO].[dbo].[RT_DA_LMP]
where [Day] >= @startdt and [Day] <= @enddt and (DATEPART(month, [Day]) = @month1 or DATEPART(month, [Day]) = @month2 or
DATEPART(month, [Day]) = @month3 or DATEPART(month, [Day]) = @month4 or DATEPART(month, [Day]) = @month5 or
DATEPART(month, [Day]) = @month6 or DATEPART(month, [Day]) = @month7 or DATEPART(month, [Day]) = @month8 or
DATEPART(month, [Day]) = @month9 or DATEPART(month, [Day]) = @month10 or DATEPART(month, [Day]) = @month11 or DATEPART(month, [Day]) = @month12)
and (DATEPART(weekday, [day]) = @day1 or DATEPART(weekday, [day]) = @day2 or DATEPART(weekday, [day]) = @day3
or DATEPART(weekday, [day]) = @day4 or DATEPART(weekday, [day]) = @day5 or DATEPART(weekday, [day]) = @day6 or DATEPART(weekday, [day]) = @day7)
and NID = @nid
select @UB = AVG([HE01])+ ((1.96)*(STDEV([HE01]))/sqrt(@samplecount))
from [NYISO].[dbo].[RT_DA_LMP]
where [Day] >= @startdt and [Day] <= @enddt and (DATEPART(month, [Day]) = @month1 or DATEPART(month, [Day]) = @month2 or
DATEPART(month, [Day]) = @month3 or DATEPART(month, [Day]) = @month4 or DATEPART(month, [Day]) = @month5 or
DATEPART(month, [Day]) = @month6 or DATEPART(month, [Day]) = @month7 or DATEPART(month, [Day]) = @month8 or
DATEPART(month, [Day]) = @month9 or DATEPART(month, [Day]) = @month10 or DATEPART(month, [Day]) = @month11 or DATEPART(month, [Day]) = @month12)
and (DATEPART(weekday, [day]) = @day1 or DATEPART(weekday, [day]) = @day2 or DATEPART(weekday, [day]) = @day3
or DATEPART(weekday, [day]) = @day4 or DATEPART(weekday, [day]) = @day5 or DATEPART(weekday, [day]) = @day6 or DATEPART(weekday, [day]) = @day7)
and NID = @nid
select @LB =AVG([HE01])- ((1.96)*(STDEV([HE01]))/sqrt(@samplecount))
from [NYISO].[dbo].[RT_DA_LMP]
where [Day] >= @startdt and [Day] <= @enddt and (DATEPART(month, [Day]) = @month1 or DATEPART(month, [Day]) = @month2 or
DATEPART(month, [Day]) = @month3 or DATEPART(month, [Day]) = @month4 or DATEPART(month, [Day]) = @month5 or
DATEPART(month, [Day]) = @month6 or DATEPART(month, [Day]) = @month7 or DATEPART(month, [Day]) = @month8 or
DATEPART(month, [Day]) = @month9 or DATEPART(month, [Day]) = @month10 or DATEPART(month, [Day]) = @month11 or DATEPART(month, [Day]) = @month12)
and (DATEPART(weekday, [day]) = @day1 or DATEPART(weekday, [day]) = @day2 or DATEPART(weekday, [day]) = @day3
or DATEPART(weekday, [day]) = @day4 or DATEPART(weekday, [day]) = @day5 or DATEPART(weekday, [day]) = @day6 or DATEPART(weekday, [day]) = @day7)
and NID = @nid
select @MErr = (1.96)*(STDEV([HE01]))/sqrt(@samplecount)
from [NYISO].[dbo].[RT_DA_LMP]
where [Day] >= @startdt and [Day] <= @enddt and (DATEPART(month, [Day]) = @month1 or DATEPART(month, [Day]) = @month2 or
DATEPART(month, [Day]) = @month3 or DATEPART(month, [Day]) = @month4 or DATEPART(month, [Day]) = @month5 or
DATEPART(month, [Day]) = @month6 or DATEPART(month, [Day]) = @month7 or DATEPART(month, [Day]) = @month8 or
DATEPART(month, [Day]) = @month9 or DATEPART(month, [Day]) = @month10 or DATEPART(month, [Day]) = @month11 or DATEPART(month, [Day]) = @month12)
and (DATEPART(weekday, [day]) = @day1 or DATEPART(weekday, [day]) = @day2 or DATEPART(weekday, [day]) = @day3
or DATEPART(weekday, [day]) = @day4 or DATEPART(weekday, [day]) = @day5 or DATEPART(weekday, [day]) = @day6 or DATEPART(weekday, [day]) = @day7)
and NID = @nid

select @RTAVG as RTAVG, @DAAVG as DAAVG, @DIFFAVG as DIFFAVG, @STDDEV as STDDEV, @UB as UpperBound, @LB as LowerBound, @MErr as MarginErr, @samplecount
```

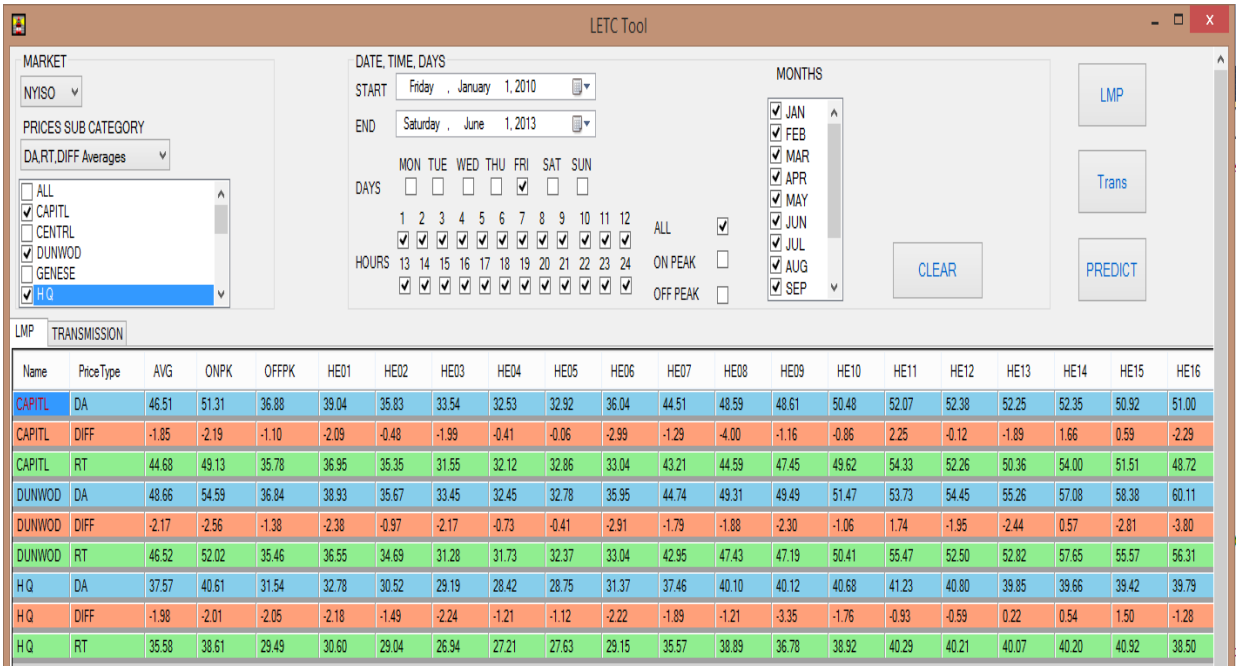
Figure 22. SQL Stored Procedure Example.

5. EXPERIMENTS

We are going to simulate a few scenarios by using the tools and calculate the risk of buying electricity for a particular zone at particular hours of the day.

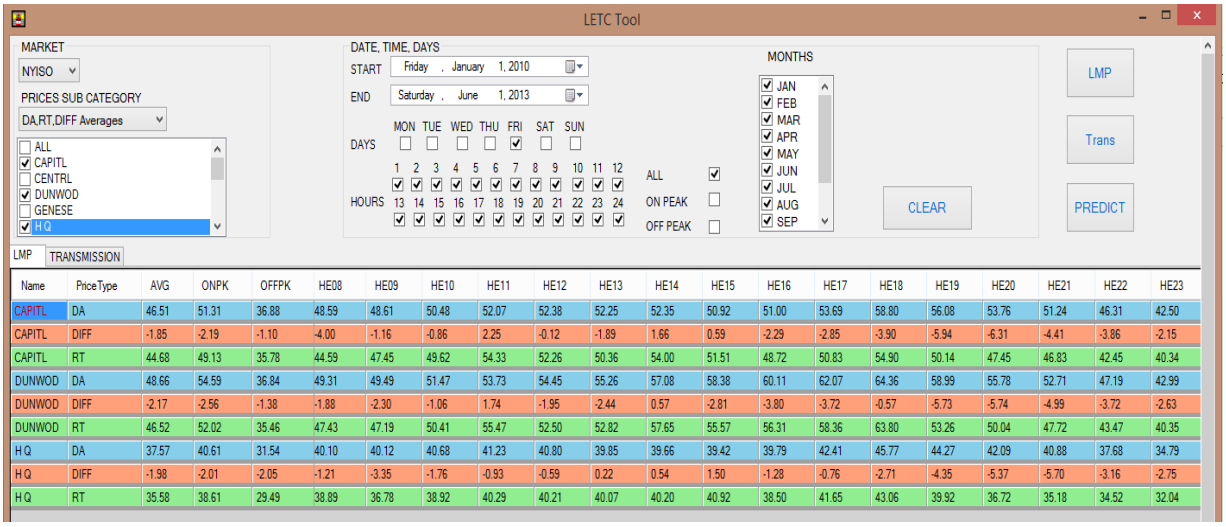
The image below shows the DA, RT, DIFF averages for all the Fridays in between January 1 2010 to June 1 2013. There are two screenshots as all the 24 hours cannot be displayed without scrolling the data grid.

Here as shown in Figure 23, the users can view the averages of price types for the selected zones.



Name	PriceType	AVG	ONPK	OFFPK	HE01	HE02	HE03	HE04	HE05	HE06	HE07	HE08	HE09	HE10	HE11	HE12	HE13	HE14	HE15	HE16
CAPITL	DA	46.51	51.31	36.88	39.04	35.83	33.54	32.53	32.92	36.04	44.51	48.59	48.61	50.48	52.07	52.38	52.25	52.35	50.92	51.00
CAPITL	DIFF	-1.85	-2.19	-1.10	-2.09	-0.48	-1.99	-0.41	-0.06	-2.99	-1.29	-4.00	-1.16	-0.86	2.25	-0.12	-1.89	1.66	0.59	-2.29
CAPITL	RT	44.68	49.13	35.78	36.95	35.35	31.55	32.12	32.86	33.04	43.21	44.59	47.45	49.62	54.33	52.26	50.36	54.00	51.51	48.72
DUNWOD	DA	48.66	54.59	36.84	38.93	35.67	33.45	32.45	32.78	35.95	44.74	49.31	49.49	51.47	53.73	54.45	55.26	57.08	58.38	60.11
DUNWOD	DIFF	-2.17	-2.56	-1.38	-2.38	-0.97	-2.17	-0.73	-0.41	-2.91	-1.79	-1.88	-2.30	-1.06	1.74	-1.95	-2.44	0.57	-2.81	-3.80
DUNWOD	RT	46.52	52.02	35.46	36.55	34.69	31.28	31.73	32.37	33.04	42.95	47.43	47.19	50.41	55.47	52.50	52.82	57.65	55.57	56.31
H Q	DA	37.57	40.61	31.54	32.78	30.52	29.19	28.42	28.75	31.37	37.46	40.10	40.12	40.68	41.23	40.80	39.85	39.66	39.42	39.79
H Q	DIFF	-1.98	-2.01	-2.05	-2.18	-1.49	-2.24	-1.21	-1.12	-2.22	-1.89	-1.21	-3.35	-1.76	-0.93	-0.59	0.22	0.54	1.50	-1.28
H Q	RT	35.58	38.61	29.49	30.60	29.04	26.94	27.21	27.63	29.15	35.57	38.89	36.78	38.92	40.29	40.21	40.07	40.20	40.92	38.50

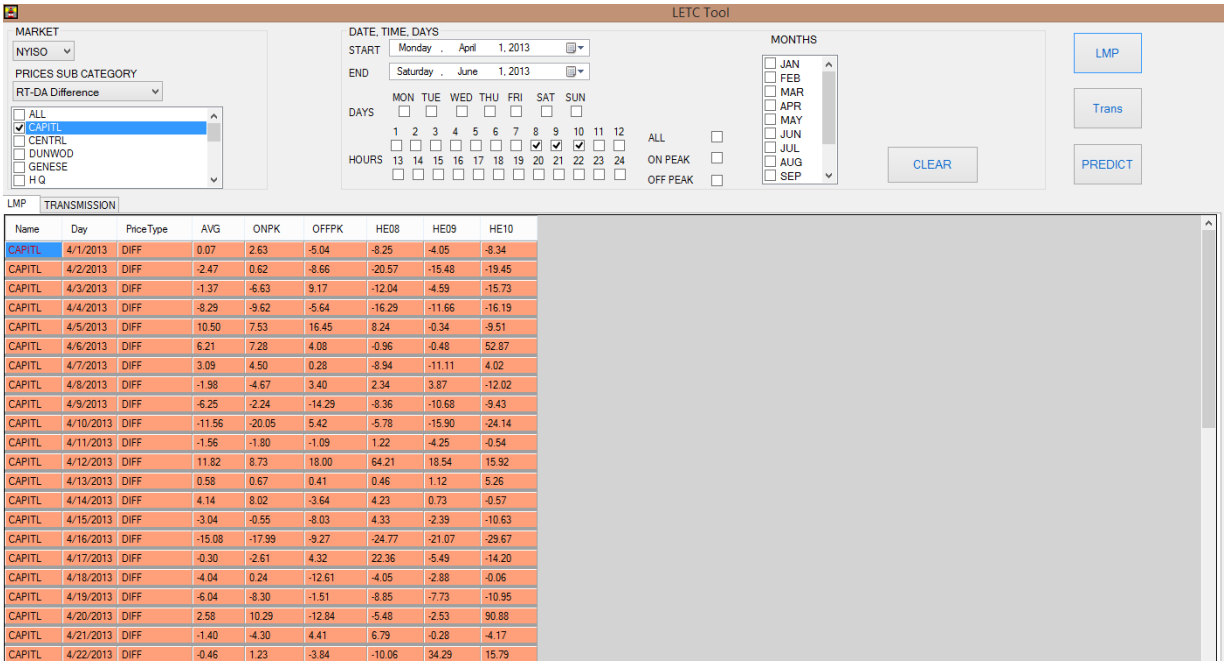
Figure 23. Average Prices Displayed on the GUI for Hours 01 to 16.



Name	PriceType	AVG	ONPK	OFFPK	HE08	HE09	HE10	HE11	HE12	HE13	HE14	HE15	HE16	HE17	HE18	HE19	HE20	HE21	HE22	HE23
CAPITL	DA	46.51	51.31	36.88	48.59	48.61	50.48	52.07	52.38	52.25	52.35	50.92	51.00	53.69	58.80	56.08	53.76	51.24	46.31	42.50
CAPITL	DIFF	-1.85	-2.19	-1.10	-4.00	-1.16	-0.86	2.25	-0.12	-1.89	1.66	0.59	-2.29	-2.85	-3.90	-5.94	-6.31	-4.41	-3.86	-2.15
CAPITL	RT	44.68	49.13	35.78	44.59	47.45	49.62	54.33	52.26	50.36	54.00	51.51	48.72	50.83	54.90	50.14	47.45	46.83	42.45	40.34
DUNWOD	DA	48.66	54.59	36.84	49.31	49.49	51.47	53.73	54.45	55.26	57.08	58.38	60.11	62.07	64.36	58.99	55.78	52.71	47.19	42.99
DUNWOD	DIFF	-2.17	-2.56	-1.38	-1.88	-2.30	-1.06	1.74	-1.95	-2.44	0.57	-2.81	-3.80	-3.72	-0.57	-5.73	-5.74	-4.99	-3.72	-2.63
DUNWOD	RT	46.52	52.02	35.46	47.43	47.19	50.41	55.47	52.50	52.82	57.65	55.57	56.31	58.36	63.80	53.26	50.04	47.72	43.47	40.35
H Q	DA	37.57	40.61	31.54	40.10	40.12	40.68	41.23	40.80	39.85	39.66	39.42	39.79	42.41	45.77	44.27	42.09	40.88	37.68	34.79
H Q	DIFF	-1.98	-2.01	-2.05	-1.21	-3.35	-1.76	-0.93	-0.59	0.22	0.54	1.50	-1.28	-0.76	-2.71	-4.35	-5.37	-5.70	-3.16	-2.75
H Q	RT	35.58	38.61	29.49	38.89	36.78	38.92	40.29	40.21	40.07	40.20	40.92	38.50	41.65	43.06	39.92	36.72	35.18	34.52	32.04

Figure 24. Average Prices Displayed on the GUI for Hours 08 to 23.

Using the average functions as shown in Figure 23 and Figure 24, we can easily spot some of the profitable hours to buy electricity based on a positive DIFF Average. And we can further drill down into the data by selecting a few hours and particular days as shown in Figure 25 and Figure 26.



Name	Day	PriceType	AVG	ONPK	OFFPK	HE08	HE09	HE10
CAPITL	4/1/2013	DIFF	0.07	2.63	-5.04	-8.25	-4.05	-8.34
CAPITL	4/2/2013	DIFF	-2.47	0.62	-8.66	-20.57	-15.48	-19.45
CAPITL	4/3/2013	DIFF	-1.37	-6.63	9.17	-12.04	-4.59	-15.73
CAPITL	4/4/2013	DIFF	-8.29	-9.62	-5.64	-16.29	-11.66	-16.19
CAPITL	4/5/2013	DIFF	10.50	7.53	16.45	8.24	-0.34	-9.51
CAPITL	4/6/2013	DIFF	6.21	7.28	4.08	-0.96	-0.48	52.87
CAPITL	4/7/2013	DIFF	3.09	4.50	0.28	-8.94	-11.11	4.02
CAPITL	4/8/2013	DIFF	-1.98	-4.67	3.40	2.34	3.87	-12.02
CAPITL	4/9/2013	DIFF	-6.25	-2.24	-14.29	-8.36	-10.68	-9.43
CAPITL	4/10/2013	DIFF	-11.56	-20.05	5.42	-5.78	-15.90	-24.14
CAPITL	4/11/2013	DIFF	-1.56	-1.80	-1.09	1.22	-4.25	-0.54
CAPITL	4/12/2013	DIFF	11.82	8.73	18.00	64.21	18.54	15.92
CAPITL	4/13/2013	DIFF	0.58	0.67	0.41	0.46	1.12	5.26
CAPITL	4/14/2013	DIFF	4.14	8.02	-3.64	4.23	0.73	-0.57
CAPITL	4/15/2013	DIFF	-3.04	-0.55	-8.03	4.33	-2.39	-10.63
CAPITL	4/16/2013	DIFF	-15.08	-17.99	-9.27	-24.77	-21.07	-29.67
CAPITL	4/17/2013	DIFF	-0.30	-2.61	4.32	22.36	-5.49	-14.20
CAPITL	4/18/2013	DIFF	-4.04	0.24	-12.61	-4.05	-2.88	-0.06
CAPITL	4/19/2013	DIFF	-6.04	-8.30	-1.51	-8.85	-7.73	-10.95
CAPITL	4/20/2013	DIFF	2.58	10.29	-12.84	-5.48	-2.53	90.88
CAPITL	4/21/2013	DIFF	-1.40	-4.30	4.41	6.79	-0.28	-4.17
CAPITL	4/22/2013	DIFF	-0.46	1.23	-3.84	-10.06	34.29	15.79

Figure 25. RT-DA Differences Displayed for Selected Inputs.

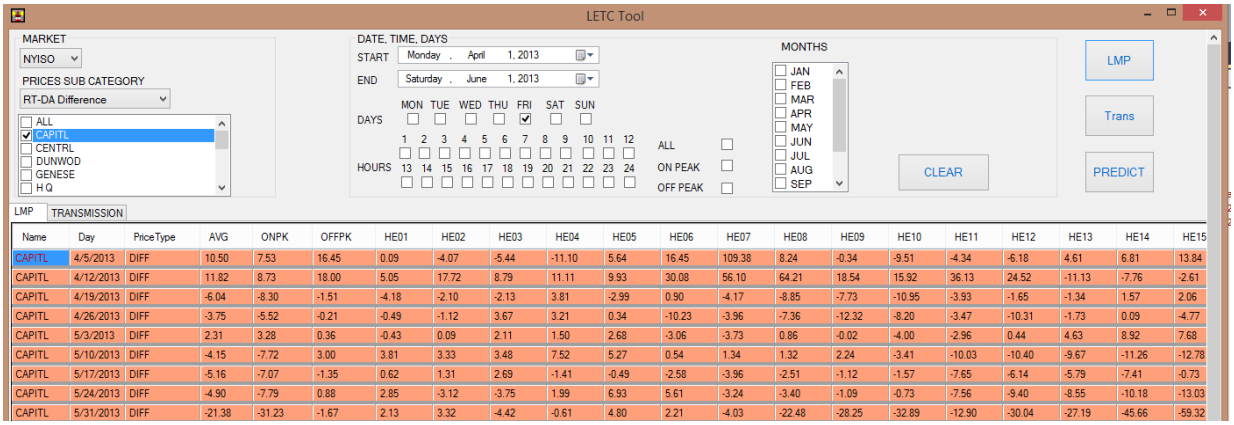


Figure 26. RT-DA Differences Displayed for Selected Inputs.

Using Prediction: The prediction window can be used separately to check for probabilities and risks of buying electricity for certain scenarios.

If a trader is interested in buying electricity for particular hours of a particular day, he can run various simulations as below to find the risk and can perform more searches on the main window for a detailed set of data.

The Figures 27, 28, 29, 30 are a few examples of different estimation simulations with different selections of hours and days and date ranges. The prediction column in the data grid displays various risks associated respectively.

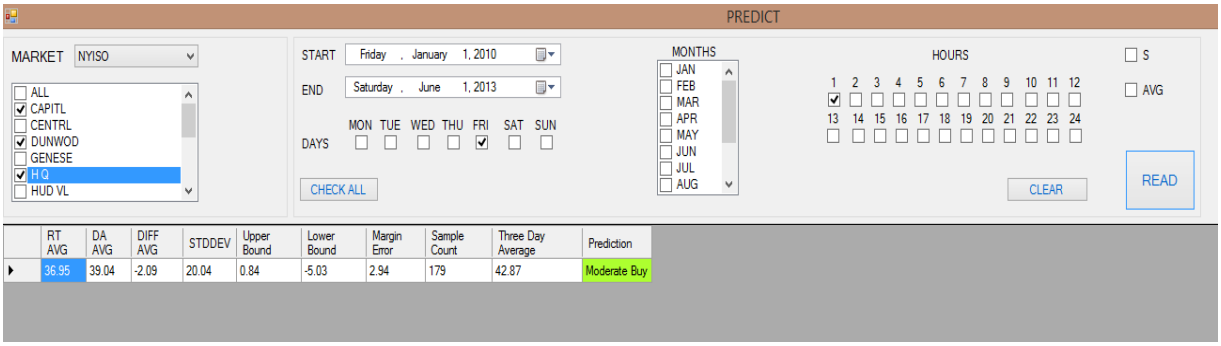


Figure 27. Estimation 1.

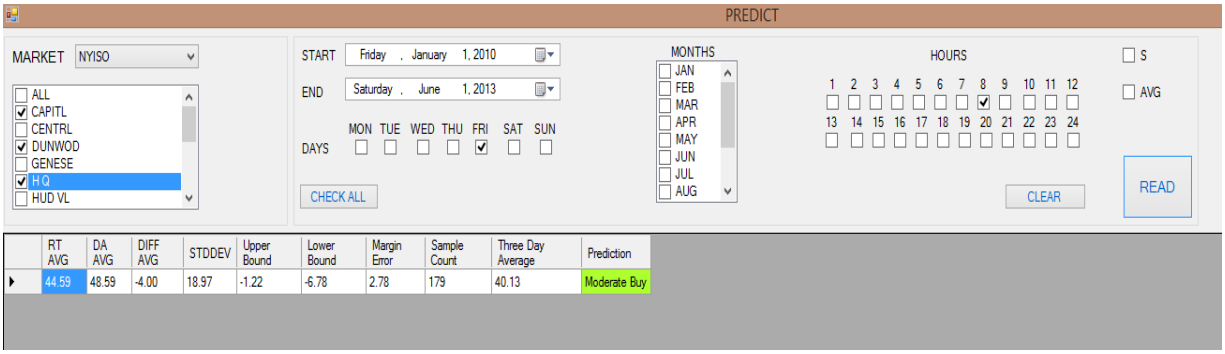


Figure 28. Estimation 2.

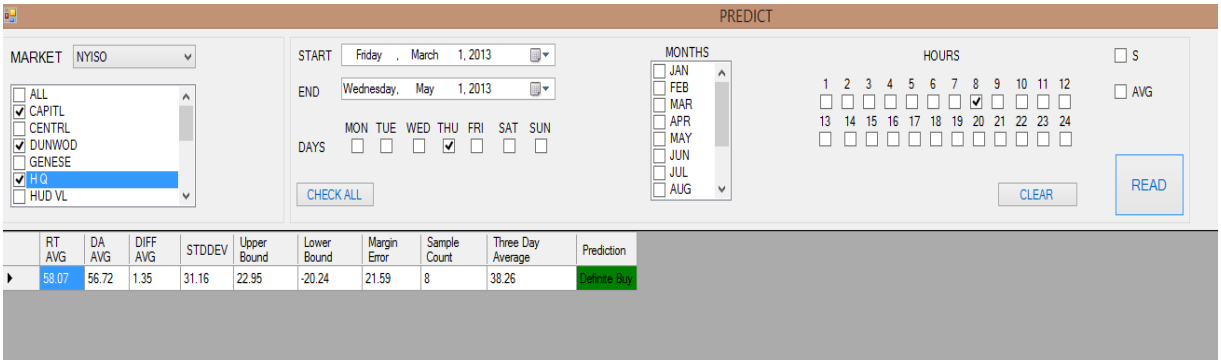


Figure 29. Estimation 3.

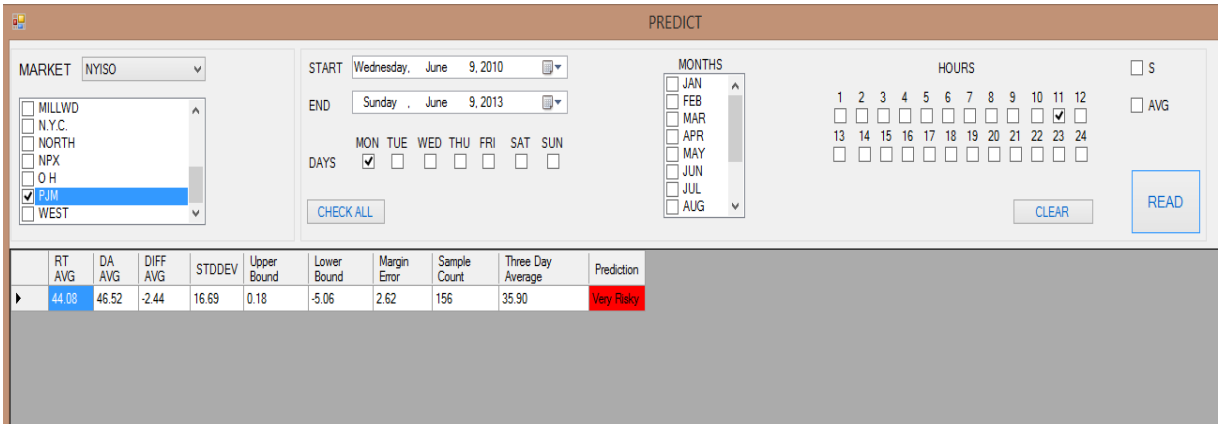


Figure 30. Estimation 4.

6. CONCLUSION, LIMITATIONS AND FUTURE WORK

6.1. Conclusion

This current research aims to provide the users with a historical analysis and estimation tool with a user friendly and efficient GUI, reduce the usage of cumbersome and time taking spreadsheets for analyzing large quantities of data as well as some basic risk prediction functionality.

The databases have been designed to perform consistently and efficiently as per the requirements of the tool and can be easily modified with changing needs. And an object oriented language has been used for coding the project in order to implement the latest programming principles with focus on efficiency, easy to test and manage if changes are required. Thus providing a good platform to invest into and develop in the future.

6.2. Future Work and Limitations

This tools efficiency and functionalities can be effectively maximized by adding the ability to have weather data, oil and natural gas prices and other constraints information. Having these additional sets of information would take a huge effort and time but can provide a great tool for traders to use and make profits.

REFERENCES

[1] *About NYISO*. (2013, 12 15). Retrieved from NYISO: http://www.nyiso.com/public/about_nyiso/nyisoataglance/index.jsp

[2] *C# Programming Language*. (2013, 12 15). Retrieved from MSDN: http://msdn.microsoft.com/en-us/library/z1zx9t92.aspx

[3] *CME Group Energy Tools*. (2013, 12 15). Retrieved from CME Group: http://www.cmegroup.com/tools-information/energy-tools.html

[4] *Confidence Interval*. (2013, 12 15). Retrieved from mathworld.wolfram.com: http://mathworld.wolfram.com/ConfidenceInterval.html

[5] *Database Design*. (2013, 11 01). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Database_design

[6] *Electricity Market*. (2013, 12 15). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Electricity_market

[7] *Energy Markets*. (2013, 12 15). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Energy_market

[8] *FERC.* (2013, 11 01). Retrieved from Energy Primer: http://www.ferc.gov/market-oversight/guide/energy-primer.pdf

[9] Heike Brand, E. T. (2002). Market analysis and tool for Electricity Trading.

[10] *ISO Maps*. (2013, 11 01). Retrieved from NYSEG: http://www.nyseg.com/SuppliersAndPartners/electricityescos/isomaps/default.html

[11] *Microsoft SQL Server*. (2013, 11 01). Retrieved from Microsoft: https://www.microsoft.com/en-us/sqlserver/product-info.aspx

[12] *Microsoft Visual Studio*. (2013, 11 01). Retrieved from MSDN: http://msdn.microsoft.com/en-us/vstudio/cc136611.aspx

[13] *NYISO Market Data*. (2013, 11 01). Retrieved from NYISO: http://www.nyiso.com/public/markets_operations/market_data/pricing_data/index.jsp

[14] *OATI Data Analytics*. (2013, 11 01). Retrieved from OATI: http://www.oati.com/Solution/Energy-Trading-RIsk-Management/Data-Analytics

[15] *OATI Risk Management*. (2013, 11 01). Retrieved from OATI: http://www.oati.com/Solution/Energy-Trading-RIsk-Management/Risk-Management

[16] *OATI Trading and Scheduling*. (2013, 11 01). Retrieved from OATI: http://www.oati.com/Solution/Energy-Trading-RIsk-Management/Trading-Scheduling

[17] *Object Oriented Programming*. (2013, 11 01). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Object-oriented_programming

[18] Singh, P. B. (2004). Electricity Trading In Competitive Power Market: An Overview And Key Issues. *INTERNATIONAL CONFERENCE ON POWER SYSTEMS, ICPS2004, KATHMANDU, NEPAL (P110)* .

[19] Spolsky, J. (2013). User Interface Design for Programmers. In J. Spolsky, *User Interface Design for Programmers* (pp. 1-60). Retrieved from Wikipedia: http://en.wikibooks.org/wiki/GUI_Design_Principles

[20] *Standard Deviation*. (2013, 11 01). Retrieved from mathworld.wolfram.com: http://mathworld.wolfram.com/StandardDeviation.html

[21] *Sungard - Marketing and Trading Tools*. (2013, 11 01). Retrieved from Sungard Financial Systems: http://financialsystems.sungard.com/solutions/energy-trading-operations/marketing-trading

[22] *TIBCO - Discover Spotfire*. (2013, 11 01). Retrieved from TIBCO: http://spotfire.tibco.com/en/discover-spotfire/who-uses-spotfire/by-industry/energy.aspx

[23] *Variance*. (2013, 11 01). Retrieved from mathworld.wolfram.com: http://mathworld.wolfram.com/Variance.html

[24] *Ventyx - Energy Market Intelligence & Forecasting Solutions*. (2013, 11 01). Retrieved from Ventyx: http://www.ventyx.com/en/enterprise/business-operations/energy-mkt-intell?kw=energy%20trading%20software&cid=701600000007MJg&gclid=CMPpo-3KkLsCFdE-MgodlTkAzA