Improving agent bidding in Power Stock Markets through a data mining enhanced agent platform

Anthony C. Chrysopoulos Andreas L Symeonidis Pericles A. Mitkas

Electrical & Computer Engineering Department

Aristotle University of Thessaloniki

GR 54124, Thessaloniki, Greece

chrysopoulos18@yahoo.gr asymeon@eng.auth.gr mitkas@eng.auth.gr

ABSTRACT

Like in any other auctioning environment, entities participating in Power Stock Markets have to compete against other in order to maximize own revenue. Towards the satisfaction of their goal, these entities (agents – human or software ones) may adopt different types of strategies – from naïve to extremely complex ones – in order to identify the most profitable goods compilation, the appropriate price to buy or sell etc, always under time pressure and auction environment constraints. Decisions become even more difficult to make in case one takes the vast volumes of historical data available into account: goods' prices, market fluctuations, bidding habits and buying opportunities. Within the context of this paper we present Cassandra, a multi-agent platform that exploits data mining, in order to extract efficient models for predicting Power Settlement prices and Power Load values in typical Day-ahead Power markets. The functionality of Cassandra is discussed, while focus is given on the bidding mechanism of Cassandra's agents, and the way data mining analysis is performed in order to generate the optimal forecasting models. Cassandra has been tested in a real-world scenario, with data derived from the Greek Energy Stock market.

1. INTRODUCTION

Agent technology has already proven its potential in various aspects of real-world trading and electronic markets. In complex environments, such as (Power) Stock Markets, where bipartite relationships between involved actors demand negotiation in order to come to an agreement, the utilization of autonomous and intelligent agents has become imminent [1, 2]. Numerous approaches have been employed in order to develop the optimal agent strategy with respect to the challenges the agents face. Among all issues rising when designing and developing systems for such highly dynamic markets, the extreme rate that data is generated at is considered of high importance. Additionally, like in all auction environments, decisions have to be made under extreme time pressure, making it difficult to apply simple algorithms and/or analytic strategies. These factors indicate that Data Mining (DM) could be a suitable technology for achieving an "intelligent" and efficient software solution.

Within the context of our work we provide a flexible, robust and powerful tool for dealing with all the issues related to the hyperactive and continuously changing Power Stock Market. We have built a multi-agent system (MAS) capable of efficiently handling the deluge of available data and of practicing various DM methodologies, in order to efficiently predict the prices of goods of interest. Our platform was benchmarked on data provided by the Greek Power Stock Market, which is a dynamic, partially observable environment. This environment allows for different strategic approaches to be followed, while it is highly challenging, due to the fact that each decision made affects instantly the future moves or decisions of the platform and the Stock Market itself.

Looking at the bigger picture, one may argue that an agent developed can employ DM, in order to extract useful nuggets of knowledge that could give him/her a predictive advantage over other competitors. In this context, we have applied DM in order to: a) analyze the historical data from the past auctions in order to predict the future values in goods of interest and, b) induce market behavior models and incorporate them into or agents, in order to provide them with a predictive edge over the competition.

The rest of the paper is organized as follows: Section 2 provides an overview of the Power Stock Market mechanisms (Auctions and Energy Market) available, as well as a state-of-the-art analysis. Section 3 presents the platform developed for monitoring and participating in the Power Stock market, and briefly discusses its architecture. Section 4 describes in detail the DM methodology applied, in order to decide on the optimal forecasting bid model, while Section 5 provides a pilot case scenario, aiming to illustrate the functionality of our platform. Finally, Section 6 summarizes work conducted and concludes the paper.

2. POWER STOCK MARKET

Up until recently, in most EU countries (Greece included), power supply was a physical monopoly, thus the establishment of a state or state-controlled administration department that would be responsible for producing, transferring and distributing, was justified. However, the advancement to more loose economic competition models has signified the cease of this physical monopoly, as far as the production and the distribution of energy are concerned.

In turn, the liberation of the Power Market has given room for the development of Open Markets, where participants are able to choose between different energy products in different periods of time and may negotiate on their "folder of products". These folders can be negotiated under three different schemes:

- **The Long-term Market**, where participants come to direct agreements in form of long-term contracts.
- **The Day-ahead Market**, where buyers place their bids in 24 hourly auctions, in order to establish a contract for the next day.

- **<u>The Real-time Market</u>**, where buyers place their bids in order to establish a contract for the next hour.

Due to the physical model of the energy production-distribution-consumption network, long-term and day-ahead markets are the most dominant ones. Nevertheless, the inability to store Power for long periods of time, dictates the development of a mechanism that can efficiently balance the supply and demand of Power and can be easily and constantly controlled. The administrator of each Power system is responsible for ensuring this balance between production and consumption, through the utilization of a Real-Time Market. The Real-Time Market model bears the highest risk for the participants, since the malfunction of an Electric node or the shutting down of a Power line can bring exaggerated rising or falling of the prices. Nevertheless, its existence is decisive for the coverage of Demand, in case the other two Markets do not provide enough Power. Additionally, one should keep in mind that high risk also implies high profit maximization for players willing to participate.

2.1 Power Market Auctions

An *auction* is defined as a strict set of rules for the specification of the exchange conditions of goods [3]. In each auction, a number of transactions are performed between the participants, where each transaction comprises two elements: a) a protocol that defines the rules of the transaction mechanism as well as the actions allowed to an (human or software) agent participating in an auction and, b) a strategy, i.e. the methodology followed by an agent in order to fulfill its goal. The protocol of an auction is determined during its design and is announced to all the participants from the beginning. Agents' strategy is designed by each participant and is unknown to the rest.

In *Power Market Auctions*, two are the most important entities:

1. The Market participants (or Players)
2. The Independent System Administrator (ISA)

A *Player* is defined as any economical entity that accesses the Power Market [4]. In general, this entity may possess a group of Production Units or/and a group of Consumers. Each Player participating in the Power Market as a *Producer* should submit his/her power supply offers in pre-specified time intervals. Each offer contains the amount of supplying Power, as well as the minimum price he/she is willing to accept. One the other hand, each Player that participates in the Power Market as a *Consumer* should submit his/her power supply demands – within the same time intervals – along with the maximum price he/she is willing to pay for it.

The *ISA* is the administrator of the Power Transfer System, and also the Administrator of the Power Stock Market. Thus, ISA is responsible for the Settlement of the Power Market, taking into consideration the transferring limitations of the system. ISA collects bids for each hourly auction and has to calculate two curves: the aggregate ascending Supply Curve and the aggregate descending Demand Curve. In the simplified case where no transfer limitations are presented, the Settlement of the Market is the intersections of the two curves (Figure 1). This point determines the *Settlement Price of the Market* (SPM) – this is the price that the Producers are paid by the Consumers, the load production of each Production Unit and the consumption of each Consumer.

In case transfer limitations exist, ISA must follow a more complicated process. He/she then has to solve an optimization problem targeting to the maximization of the social prosperity (target function). Then the Power price, the Load production of each Production Unit and the consumption of each Consumer are calculated.

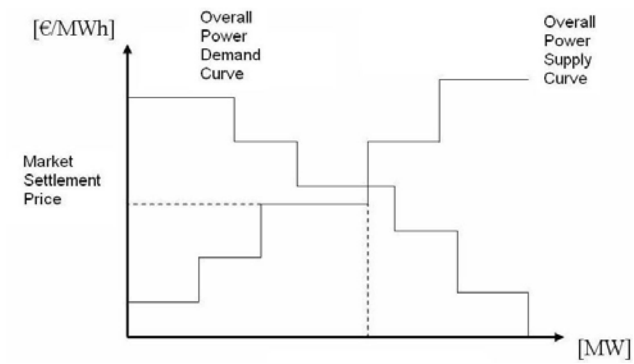


Figure 1. The aggregate Power Supply and Demand Curves

2.2 State-of-the-art

Various approaches have been employed for analyzing the behavior of Power Markets, some of which have adopted AT and DM primitives. In fact, results reported seem quite promising.

The Electric Power research Institute (ERPI) developed SEPIA (Simulator for Electrical Power Industry Agents), a multi-agent platform capable of running a plethora of computing experiments for many different market scenarios [5, 6]. SEPIA employs two different options for agent learning: a variation of the Q-Learning Algorithm [7], which corresponds each noticeable state to a suitable action, and an LCS (Learning Classifier System) morph [8], which utilizes a rule-based model and agents learn through amplified learning and genetic algorithms.

The Argonne National Laboratory, on the other hand, developed EMCAS (Electricity Market Complex Adaptive System) [9], an efficient implementation for handling the Electric Energy Market. Through EMCAS, one may study the complex interactions between the physical entities of the market, in order to analyze the participants and their strategies. Players' learning is based on genetic algorithms, while EMCAS supports stock market transactions, as well as bipartite contracts.

As far as the real-market analysis is concerned, Bagnall [10] has presented a simplified simulation model of Great Britain's Electricity Market, were the Producers are agents that participate in a series of auctions–games. In any phase of the process, a Producer is faced with two options: a) try an already tested and applicable strategy that will reassure he/she will not lose money or, b) find a new strategy rule that will maximize his/her profits. To this end, each agent is modeled with LCS learning abilities and its behavior is monitored. Changes in the behavior are caused by the transition from a uniform pricing system to a system where each producer is paid at his/her supply price. Additionally, the possibility of cooperation between two agents when the rest Producers make offers in their cost limit is also observed.

Finally, Petrov and Sheble [11] introduced Genetic Programming in their simulation and tried to model the bipartite Auctions for the Electric Energy Market, by the use of agents.

One of the players incorporates knowledge represented as a Decision-Making Tree, which is developed by Genetic Programming. The rest of the agent–players incorporate ruled–defined behaviors.

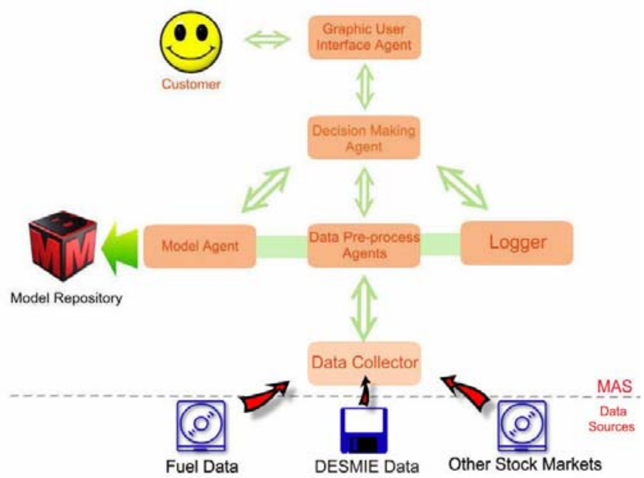


Figure 2. The Cassandra 4-layered architecture

3. DEVELOPED SYSTEM

Cassandra is a multi-agent platform designed and developed to function in an automatic and semi-autonomous manner in Energy Markets. Cassandra employs DM techniques in order to forecast the Settlement Prices, as well as the power load of the Day-Ahead Market. The implemented system may even function in a fully autonomous manner if granted permission, and may proceed with the necessary actions for the establishing a contract. The efficiency of the system is highly dependent on the models generated from historical data, as well as a set of the fail-safe rule base specified by the Power Market expert. Through Cassandra's interface, DM models are re-built dynamically, in order to study, evaluate and compare the results and choose the one that optimally projects running market trends.

3.1 Cassandra Architecture

Cassandra follows the IRF Architecture Model (Intelligent Recommendation Framework) [12], which defines a 4-layer functional model for the agent system. IRF is usually employed in enterprises for the optimization of the administration mechanism, since it can automate and integrate all the data producing or data demanding facets of a company.

Taking a closer look of the Power Market through the IRF prism, one may identify several tasks that have to be tackled:

- Collection of the historical data from previous auctions and processing
- Application of the suitable DM algorithms in order to build the necessary forecasting models
- Integration of generated models in the Business Intelligence of the System and evaluation of the results.
- Continuous monitoring Stock Market.

As expected, Cassandra employs a modular architecture (Figure 2), where each module is responsible for one of the aforementioned tasks. The platform also provides a wrapper around all modules and ensures communication with the system users. The modules comprising Cassandra are:

i. **Data Collection Module (DCM):** It is responsible for the collection of historical data, either from files provided by the user, or directly from the Internet.
ii. **Data Processing and Mining Module (DPMM):** One of the core modules of Cassandra, which is responsible for the processing of the data, preparing the training sets and applying the DM algorithms.
iii. **Decision Making Module (DMM):** It aggregates all information in order to make the optimal decision in any given occasion.
iv. **Graphic User Interface Module (GUIM):** It interacts with the users of the System. It must be user-friendly, and easily comprehensive.

3.2 Cassandra Users

Cassandra identifies three types of users:

a. System Analyst

The *System Analyst* (SA) is an expert user. He/she is the creator of the system, so he/she has absolute knowledge over it. SA is responsible for the smooth operation (unimpeded operation of the system, satisfaction of the software requirements of the users), as well as its efficiency. SA also provides domain knowledge, while he/she is responsible for generating the DM models, in order to decide on the best-performing one(s). Finally, SA checks the decision routines, in order to trace in time any errors that may occur.

b. System Manager

The *System Manager* (SM) is also an expert user, not on agents and DM, but in the Power Stock Market. SM cannot interfere directly with the system, but using his/her experience in the Market SM can easily evaluate the strategic moves of the system and decide whether Cassandra operates efficiently enough or not. SM has the authority to manually override the prediction system (for example change the bids on the Settlement Price), but he/she cannot change the models used by the system for prediction (that is under the Analyst's jurisdiction).

Table 1. Cassandra functionality

User Type	Views	Privileges	Functionality
MU	- Monitoring	None	- Account Checking - Market Monitoring - www/RSS Browsing - Chart Viewing
SM	- Monitoring - Manager	Managing only	*All the above plus:* - Manual Bidding - Logger Handling
SA	- Monitoring - Manager - Analyst	All	*All the above plus:* - Agent Overview - Model Retraining - Model Configuration - Model Statistics - Cost Evaluator - Data File Import

c. Monitoring User

The *Monitoring User* (MU) is the naïve user of the system. MU only monitors the Power Market moves and the logs the decisions – Cassandra's placed bids. In case MU notices something 'out of the ordinary' (actions that do not have the expected results), MU notifies the SM. SM must then double check MU's observations and, in case of error, notify the SA, who will react accordingly in order to optimize system operation.

Each user group is accommodated through different views of the system, enabled upon user authentication. Table 1 summarizes the functionality provided to each user group.

3.3 Implementation

Cassandra provides a multi-functional user interface to facilitate its usage. It has been implemented in Java 1.6 and all agents are developed over the Java Agent Development Framework (JADE) [13], which conforms to the FIPA specifications [14]. The system provides authorized users (SAs) full control over the agents, from their creation till their termination. All DM models are built and evaluated on the WEKA (Waikato Environment for Knowledge Analysis) API [15].

a

b

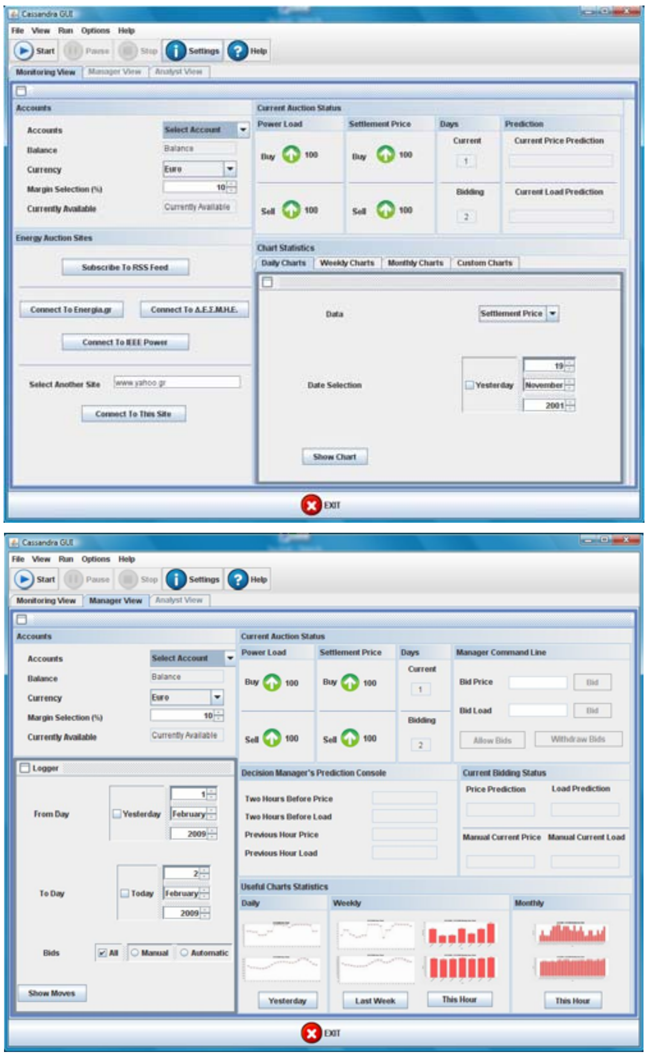


Figure 3. An overview of the Cassandra MU (a) and SM (b) View

As already denoted, Cassandra supports MUs, SMs, and SAs through respective views. When initialized, Cassandra projects the *MU View* (Figure3a). The user has then the option to log in as an SM or an SA (through the 'Settings' menu) and enjoy the privileges of each user category. The *SM View* (Figure3b) provides many additional features in comparison to the *MU View*. It realizes the *Logger*, a mechanism that provides a detailed overview of the bidding agent's predictions and actions, in order to monitor and evaluate the efficiency of the system. It also provides an *override* mechanism for manually bypassing the automated bidding process, in case SM considers the prediction to be faulty. On the other hand, the *SA View* comprises several useful design and development tools. Such tools are the *Cost Evaluator*, which calculates the cost of Production given the right Parameters, as well as a *Agent* and a *Data Mining Configuration pane* (Figure 4a and 4b, respectively), that provide full control on the MAS architecture (change the types and number of agents residing in the Processing and DM Layer, select different preprocessing agent behaviors) and the creation and evaluation of DM models (select new training and testing datasets, new algorithms/algorithm parameters etc), respectively.

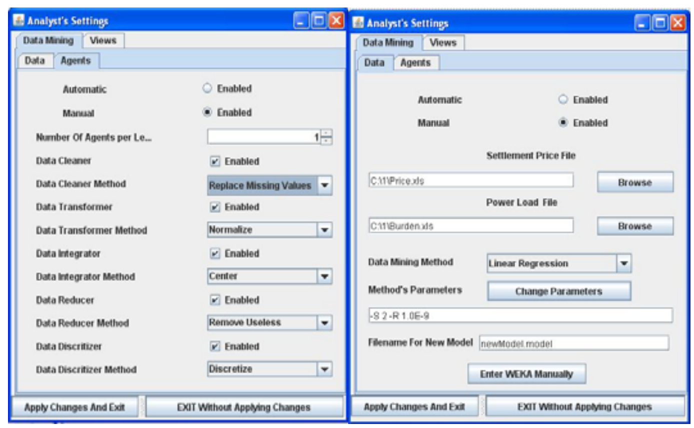


Figure 4. The Agent and Data Mining Configuration panes

4. PRELIMINARY RESULTS

Before building the Cassandra system, we performed a thorough analysis on available data, in order to build DM models that would efficiently predict the settlement price and the power load in a Day-ahead market. Both problems were modeled as regression problems, where the desired output would be the Decision Manager's prediction of the Settlement Price or Power Load with respect to past auction data. Various experiments and models were built, taking hourly/weekly/monthly periodicity into account. For the shake of simplicity, we provide the results on the models created based on a daily time window (the prior 23 hours-auctions are considered as input, requesting to predict the Settlement Prices and Power Loads for the 24th hour). The WEKA suite was employed for the conduction of the experiments with a plethora of algorithms over the available datasets.

4.1 Training

Five different classification (regression) schemes were applied, in order to decide on the one that optimally meets the problem of predicting the Settlement Price and the Power Load for the 24th hour: a) Simple Linear Regression, b) Linear

Regression, c) Isotonic Regression, d) Pace Regression, and e) Additive Regression [16].

The *Simple Linear Regression* models applied, as expected, gave poor results on every dataset it was applied on, since it is known to a very provincial technique. The *correlation coefficiency* (cc) of the model extracted was ~0.95 and ~0.98, while the *Relative Absolute Error* (RAE) was ~26% and ~22% for the Settlement Price and the Power Load, respectively. A large number of experiments were conducted applying *Linear Regression*, on different datasets (varying size) and with different algorithm parameters. The *cc* was a slightly better (0.96 and 0.98-1), as well as the *RAE* for the Settlement Price (24%). The *RAE for the* Power Load (10%), though, improved significantly. With the application of *Isotonic Regression* techniques, the results were as disappointing as in the case of *Simple Linear Regression*, only with a slight improvement in *RAE* for the Settlement Price (24%). The *Pace Regression* algorithm (numerous parameters for algorithm fine-tuning), came up with very good results both for *cc* (0.96, 1) and for *RAE* (24%, 10%) for both goods. It's should be mentioned that increasing or decreasing the *Estimator's* parameters didn't significantly influence the resulting efficiency, while in some cases the *Mean Squared Error* was much higher that the simple techniques used, indicating overfitting.

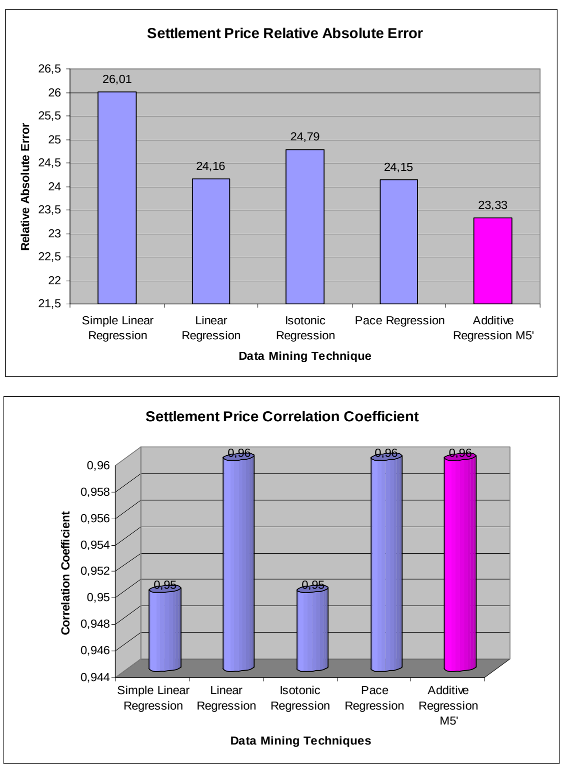


Figure 5. A comparison of the regression schemes applied on the Settlement price

4.2 Meta Classification

Apart from the regression schemas applied, we also tested some meta-classifier schemas, striving for optimal performance. Various *Additive Regression* schemes were tested against the same datasets, in order to ensure equally compared test results.

Three schemes applied to the dataset: a) *DecisionStamp* (building and using decision stamp), *REPTree* (fast decision tree learner) and *M5'* classifiers. The first performed worse than any other model extracted (*cc* 0.93 and 0.96, *RAE* 33% and 31%, respectively). Nevertheless, the *REPTree* schema came up with much better results (*cc* was 0.95 and 0.99, while *RAE* was improved to 23.21% and 14.93%, respectively). Finally, *M5'* outperformed all schemes and algorithms, with *cc* reaching the maximum value of 0.96 and 1, while in the same time *RAE* was ~23% and ~9%, respectively.

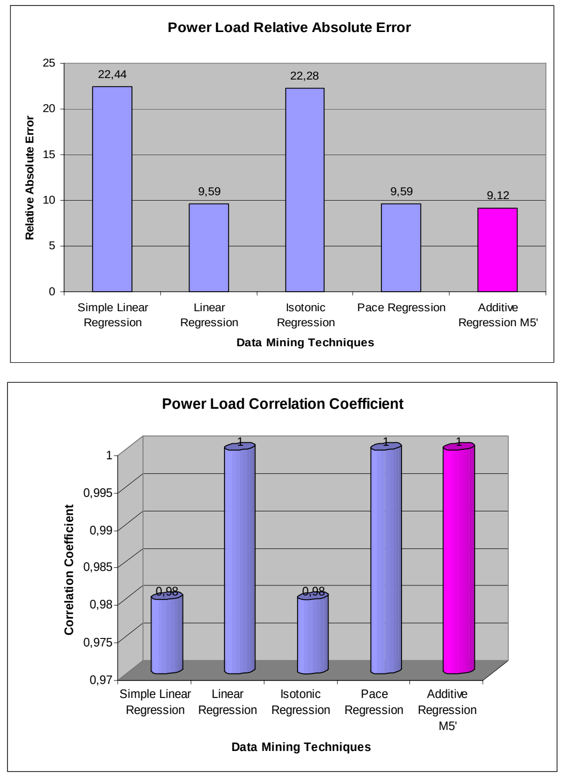


Figure 6. A comparison of the regression schemes applied on the Power load

4.3 Data Mining Experiment Results

It's obvious that the combination of the *Additive Regression* meta-classification schema with the *M5'* regression algorithm, applied on the given datasets significantly outperforms all the other learning methods applied. The main advantage of *M5'* with respect to the other methods applied is that it produced a simple and compact tree model, in contrast with *Simple* and plain *Linear Regression* that attempted to impose a linear relationship on the data, *Isotonic Regression* that picks the attribute that results the lowest square error, and *Pace Regression* that is optimal when the number of coefficients of the linear model tends to infinity. Combined with *Additive Regression*, which introduces a stochastic factor, *M5'* succeeded in improving the accuracy of the predictors and achieved optimal performance. Figures 5 and 6 provide a qualitative comparison between the regression schemes applied, cross-validated on the same dataset.

5. PILOT CASE SCENARIO

In order to demonstrate the proper functioning of Cassandra, we have set up a real-life scenario, based on historical data for the Greek Energy Stock Market (avail: `http://www.desmie.gr/`).

Within this Day-ahead market, *Producers* place each day 24-bid bundles for the forthcoming day’s 24-hourly auctions, while *Consumers* declare their needs in Power Supply, along with the maximum price willing to pay. The ISA takes all this information in consideration and calculates the aggregate Supply and Demand Curves, thus defining the *Settlement Price* and *Power Load* for each one of the forthcoming day’s auctions. All *Producers* that have made bids, less or equal than the specified Settlement Price are included in the next day’s distribution network, and are paid at the Settlement Price for each MWh sold. The rest of the *Producers* are not included in the transaction.

The scenario Cassandra is tested against is the following: First, we simulate a power stock market by randomly select 5 days from the historical data on previous auctions. For each day, Cassandra’s agents try to predict the 24th-hour auction Settlement Price and Power Load, based on the values of the other 23-hour auctions. Then the predicted values are compared to the actual ones residing in the dataset. In case the predicted price is equal or less than the actual price, we consider bidding to be successful, (Within Market ranges) for that hourly auction. If not, we consider it unsuccessful. Three experiments were conducted, where Cassandra agents employed three different DM models:

1. The first DM model was extracted by the use of Simple Linear regression on raw data (no pre-processing was performed). Figure 7a depicts the results, where one may notice that all predictions were ‘Unsuccessful’.

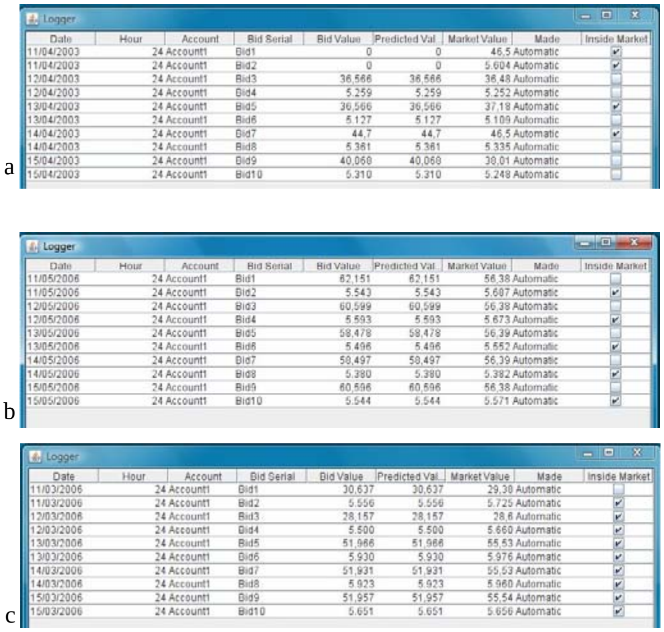
2. The second DM model was again extracted by the use of Simple Linear regression, this time on a filtered (preprocessed) dataset. Figure 7b depicts the results, where improvement can be seen.

3. Finally, the third DM model was extracted by the use of the *Additive Regression* with the *M5’* regression scheme. Figure 7c depicts the results, where improvement is obvious.

6. FUTURE WORK

Within the context of this paper we have presented Cassandra, a multi-agent system that employs DM primitives in order to automate the process of participating in the Power Stock Market. Cassandra succeeds in predicting forthcoming Settlement Prices and Power Load values, allowing its ‘master’ to bid within market ranges and maximize profit. The tool provides the ability to design the MAS and decide on the technique and algorithm on which to build the DM model on, while it provides a number of utilities for monitoring the market and analyzing trends.

Future work is focused on two directions: a) extensively study periodicity (same day each month, same weekend each year and so on), in order to identify an even more efficient model for staying ‘within market ranges’ and b) identify the maximum price to bid so as to maximize revenue. Additionally, one may work towards improving the GUI (Graphical User Interface) of the platform. The addition of more graphs, tables and diagrams would help in extracting useful information produced during the system’s operation.



Logger

Date	Hour	Account	Bid Serial	Bid Value	Predicted Val	Market Value	Made	Inside Market
11/04/2003	24	Account1	Bid1	0	0	46,5	Automatic	✔
11/04/2003	24	Account1	Bid2	0	0	5.604	Automatic	✔
12/04/2003	24	Account1	Bid3	36,566	36,566	36,48	Automatic	
12/04/2003	24	Account1	Bid4	5.259	5.259	5.252	Automatic	
13/04/2003	24	Account1	Bid5	36,566	36,566	37,18	Automatic	✔
13/04/2003	24	Account1	Bid6	5.127	5.127	5.109	Automatic	
14/04/2003	24	Account1	Bid7	44,7	44,7	46,5	Automatic	✔
14/04/2003	24	Account1	Bid8	5.361	5.361	5.335	Automatic	
15/04/2003	24	Account1	Bid9	40,060	40,060	38,01	Automatic	
15/04/2003	24	Account1	Bid10	5.310	5.310	5.248	Automatic	

a

Logger

Date	Hour	Account	Bid Serial	Bid Value	Predicted Val	Market Value	Made	Inside Market
11/05/2006	24	Account1	Bid1	62,151	62,151	56,38	Automatic	
11/05/2006	24	Account1	Bid2	5.543	5.543	5.607	Automatic	✔
12/05/2006	24	Account1	Bid3	60,599	60,599	56,38	Automatic	
12/05/2006	24	Account1	Bid4	5.593	5.593	5.673	Automatic	✔
13/05/2006	24	Account1	Bid5	58,478	58,478	56,39	Automatic	
13/05/2006	24	Account1	Bid6	5.496	5.496	5.552	Automatic	✔
14/05/2006	24	Account1	Bid7	58,497	58,497	56,39	Automatic	
14/05/2006	24	Account1	Bid8	5.380	5.380	5.382	Automatic	✔
15/05/2006	24	Account1	Bid9	60,596	60,596	56,38	Automatic	
15/05/2006	24	Account1	Bid10	5.544	5.544	5.571	Automatic	✔

b

Logger

Date	Hour	Account	Bid Serial	Bid Value	Predicted Val	Market Value	Made	Inside Market
11/03/2006	24	Account1	Bid1	30,637	30,637	29,30	Automatic	
11/03/2006	24	Account1	Bid2	5.556	5.556	5.725	Automatic	✔
12/03/2006	24	Account1	Bid3	28,157	28,157	28,6	Automatic	✔
12/03/2006	24	Account1	Bid4	5.500	5.500	5.660	Automatic	✔
13/03/2006	24	Account1	Bid5	51,966	51,966	55,53	Automatic	✔
13/03/2006	24	Account1	Bid6	5.930	5.930	5.976	Automatic	✔
14/03/2006	24	Account1	Bid7	51,931	51,931	55,53	Automatic	✔
14/03/2006	24	Account1	Bid8	5.923	5.923	5.960	Automatic	✔
15/03/2006	24	Account1	Bid9	51,957	51,957	55,54	Automatic	✔
15/03/2006	24	Account1	Bid10	5.651	5.651	5.656	Automatic	✔

c

Figure 7. The accuracy of the three applied models

7. REFERENCES

[1] M. He, N. R. Jennings, and H.-F. Leung. On Agent-Mediated Electronic Commerce. IEEE Transactions on Knowledge and Data Engineering, 15(4):985–1003, Jul/Aug 2003.

[2] P. Stone. Learning and Multiagent Reasoning for Autonomous Agents. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, pages 13–30, January 2007. IJCAI-07 Computers and Thought Paper.

[3] P.R. Wurman, M. Wellman and W.W. “The Michigan Internet Auction-bot: A Configurable Auction Server for Human and Software Agents”, Proceedings of the Second International Conference on Autonomous Agents, 1998.

[4] A.C. Tellidou, A.G. Bakirtzis, “Multi-Agent Reinforcement Learning for Strategic Bidding in Power Markets”, 3rd International IEEE Conference on Intelligent Systems, London UK, 2006.

[5] M. Amin, D. Ballard, “Defining New Markets for Intelligent Agents”, IEEE IT Professional, 2000.

[6] M. Amin, “Restructuring the Electric Enterprise ‘Simulating the Evolution of the Electric Power Industry with Intelligent Agents’ ”, Chapter 3 in “Market Analysis and Resource Management” by A. Faruqui and K. Eakin, Kluwer Publishers, 2002.

[7] Watkins, C.J.C.H. (1989).Learning from Delayed Rewards. PhD thesis, Cambridge University, Cambridge, England.

[8] John Henry Holland (1989), "Using Classifier Systems to Study Adaptive Nonlinear Networks", Lectures in the Sciences of Complexity. Addison Wesley. 1989.

[9] G. Conzelmann , G. Boyd , V. Koritarov , T. Veselka, “Multi-Agent Power Market Simulation using EMCAS” , Power Engineering Society General Meeting , 2005.

[10] A. J. Bagnall, G. D. Smith, “Game Playing with Autonomous Adaptive Agents in a Simplified Economic Model of the UK Market in Electricity Generation”, IEEE Power System Technology, 2000.

[11] V. Petrov , G. Sheble , “Power Auctions Bid Generation with Adaptive Agents using Genetic Programming” , Department of Electrical and Computer Engineering , Iowa USA , 2000.

[12] A. L. Symeonidis, P.A. Mitkas, “Agent Intelligence through Data Mining”, Gerard Weiss, 2005.

[13] F. Bellifemine, A. Poggi, and R. Rimassa. Developing multiagent systems with JADE. Lecture Notes in Computer Science,1986:89–101, 2001.

[14] T. F. Foundations. Foundation for intelligent physical agents specifications. Technical report, The FIPA Consortium, 2003.

[15] http://www.cs.waikato.ac.nz/ml/weka/

[16] David A. Freedman, Statistical Models: Theory and Practice, Cambridge University Press, 2005.