Nonlinear Optimization under Uncertainty for Sustainable Energy Informatics Problems

A Dissertation

Presented to

the Faculty of the Department of Industrial Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Industrial Engineering

by

MohammadHossein Poursaeidi

May 2013

Nonlinear Optimization under Uncertainty for Sustainable Energy Informatics Problems

MohammadHossein Poursaeidi

Approved:

Chairman of the Committee
Erhun Kundakcioglu, Assistant Professor,
Industrial Engineering

Committee Members:

Gino Lim, Associate Professor,
Industrial Engineering

Qianmei Feng, Associate Professor,
Industrial Engineering

Shishir Shah, Associate Professor,
Computer Science

Bora Gencturk, Assistant Professor,
Civil and Environmental Engineering

Suresh K. Khator, Associate Dean
Cullen College of Engineering

Gino Lim, Associate Professor and
Chairman, Industrial Engineering

Acknowledgements

This dissertation is the result of a great deal of work on the part of numerous individuals who made my years at the University of Houston a rewarding experience. The best and worst moments of my doctoral dissertation journey have been shared with many people. It has been a great privilege to spend several years in the Department of Industrial Engineering, and its members will always remain dear to me.

I wish to take this opportunity to express my most profound gratitude for my dissertation supervisor, Dr. Erhun Kundakcioglu, for giving me the opportunity and freedom to pursue my research under his supervision. He has been instrumental during my studies in the University of Houston and has provided support during my stay here. He patiently provided the vision, encouragement and advice necessary for me to proceed through the doctoral program and complete my dissertation. His profound knowledge and scientific curiosity have set high standards and are a constant source of inspiration. Interacting with him as his student and advisee, I benefited tremendously from his numerous qualities of a mentor and collaborator. His clarity of thought and keen insight have greatly influenced my research. His flexibility in scheduling, gentle encouragement and relaxed demeanor made for a good working relationship and the impetus for me to finish this research. His advice, both scholarly and non-academic, and most of all his friendship, leave me greatly indebted.

I also wish to thank the other members of my dissertation committee, Dr. Gino Lim, Dr. Qianmei Feng, Dr. Shishir Shah and Dr. Bora Gencturk for having agreed to take the time out of their busy schedules to read my manuscript and to provide me with their comments and suggestions. I have benefited greatly from the generosity and support of many faculty members and numerous friends and colleagues at the University of Houston. There are numerous friends who contribute to the excellent

working environment and creative atmosphere. I am very happy that, in many cases, my friendships with them have extended well beyond our shared time in the program. I will always appreciate their support, friendship, and love.

Last, and most importantly, I am forever indebted to my parents, Sadegh Poursaeidi and Hamideh Modarresi, my brothers, Mahmoud, Mohsen, and Mahdi and my sisters, Shokoufeh and Saeideh. Their invaluable and relentless support, encouragement, and love are without doubt the most important reasons for my success. I could not have achieved this without their unlimited sacrifice. My parents bore me, raised me, supported me, taught me, and loved me. To them and my family I dedicate this dissertation.

To my parents

Sadegh Poursaeidi & Hamideh Modarresi

with all my love

Nonlinear Optimization under Uncertainty for Sustainable Energy Informatics Problems

An Abstract

of a

Dissertation

Presented to

the Faculty of the Department of Industrial Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Industrial Engineering

by

MohammadHossein Poursaeidi

May 2013

Abstract

Despite the undeniable importance of energy in the modern world, the majority of today's energy sources are unsustainable which has environmental drawbacks such as global climate warming. Increasing sustainable energy efficiency through optimization of resources has become one of the major goals of the century due to the potential economical and environmental benefits. The analysis, design, implementation, and use of computer science models for developing energy efficient management plans are referred to as *sustainable energy informatics*. In this dissertation, three optimization and data mining approaches for sustainable energy applications are proposed. These problems deal with analyzing data under uncertainty to make a robust and reliable decision.

The first approach presents the multiple instance classification problem with application in wind farm site locating. Hard margin loss formulations that minimize the number of misclassified instances are proposed to model more robust representations of outliers. Although the problem is $\mathcal{NP}$-hard, medium sized problems can be solved to optimality in reasonable time using integer programming and constraint programming formulations. For larger problems a three phase heuristic algorithm is proposed which is shown to have superior generalization performance compared to other approaches.

Second, a layout optimization framework for offshore wind farms is proposed under widely accepted assumptions. Although wind has less environmental impact than conventional sources, onshore wind farms currently supply only 3% of the nation's electricity while reducing carbon emissions by 2.5%. Due to higher wind speeds off the coast, offshore wind farms' potential for electricity production is typically higher than onshore counterparts yet relatively more expensive to construct, operate, and

maintain. We present a rigorous mathematical model that would minimize the cost of wind energy by examining the trade-off between the advantages of packing the turbines closer together and the loss generated by wake effects.

The purpose of the last approach is to analyze historical information on the variables that potentially have a high impact on a response variable. The goal of this study is to filter out the noise using the common ground information. Considering monthly natural gas prices, we highlight the strength of a forecasting scheme through the simultaneous selection of instances and features.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

In times of environmental catastrophes and increasing energy needs, the calls for clean, cheap, and sustainable power sources are getting louder and more demanding. Although sustainable energy can produce a level of pollution to some extent, it has less environmental drawbacks such as global climate change due to carbon emission, air pollution, large freshwater usage, acid rain, and radioactive waste than other energy sources. Motivated by this fact, in 2010, the United States invested nearly $243 billion on developing sustainable energy technologies. In the light of these, it is a fact that customer demand for sustainable energy is increasing. According to a Natural Marketing Institute (NMI) survey, 55% of American consumers want companies to increase their use of sustainable energy.

Increasing sustainable energy efficiency through optimization of resources has become one of the major goals of the century for engineers due to its potential economical and environmental benefits. To achieve this goal, computer models and simulation techniques undoubtedly play an important role. Sustainable energy informatics can be defined as the analysis, design, implementation, and use of computer science models for developing appropriate management plans that increase the energy efficiency while protecting the environment. In this dissertation, we propose three data mining and optimization approaches that can be used for sustainable energy applications. Each approach introduced in this introduction is explained in detail in Chapters 2, 4 and 3.

1.1 Robust Support Vector Machines for Multiple Instance Learning

Multiple Instance Learning (MIL) is a supervised machine learning problem, where class labels are defined on the sets, referred to as *bags*, instead of individual data instances. Each instance in a negative bag is negative, whereas positive bags may contain false positives. This notion of *bags* makes multiple instance learning particularly useful for numerous interesting applications. For instance, in drug activity prediction, unless there is at least one effective ingredient (*actual positive instance*), a drug (*bag*) is ineffective (*negative labeled*). Similarly, in molecular activity prediction, in order to observe a particular activity (*positive labeled*) for a molecule (*bag*), there has to be at least one conformation (*instance*) that exhibits the desired behavior (*actual positive*). Text categorization deals with matching a document (*bag*) with a topic of interest (*positive label*) based on a set of keywords that have been frequently used in the same concept (*actual positive instances*). In image retrieval, pictures with an object of interest (*positive labeled bags*) are not expected to include that object in all segments, but only in subsets (*actual positive instances*). Image retrieval has many usages including image preprocessing in shallow water depth retrieval that can be used to find potential places for building offshore structures. For example, an image for deep water can be deceiving if there are shallow parts in the picture. So we can assume deep is positive class. If there is one segment of the picture that is deep, we assume it is deep and good for building structures although there are shallow places (which may even be better). But if it is all shallow, that implies the place is likely to take less wind.

A robust approach for MIL based on hard margin Support Vector Machine (SVM) formulations is presented in this dissertation. Our approach uses hard margin loss function. Several IP and CP formulations are developed and compared in terms of

time performance and a three-phase heuristic algorithm is developed to be used for large scale date sets. Cross validation results show that our approach provides more accurate predictions than a traditional SVM approach to MIL.

We used offshore wind farm site location data to show the implementation of our method. To find a location to build a wind farm a decision maker will use different variables like wind speed, wind availability, water temperature, depth of water, pressure, precipitation, wave speed, wave height, and distance to shore. In different locations of a site these variables are different and a decision maker may decide to invest on that site based on the overall output of our model.

1.2 Offshore Wind Farm Layout Optimization

As energy consumption across the globe continues to increase, non-renewable energy sources attract both economic and environmental concerns. These concerns create a strong motivation for researchers to improve upon the renewable energy production methods currently available. Approximately 82% of the United States' energy in 2010 was provided by fossil fuels, while only 8% came from renewable sources [88]. Wind energy represents an important renewable energy resource, as wind turbines do not produce CO_2 emissions, and are entangled with few other environmental or social concerns. With the relative abundance of wind as an energy resource and a short list of side effects, wind energy itself is seemingly one of the most important investments that will be made in renewable energy production in the near future.

Two main forms of wind energy production include onshore and offshore wind farms. With steadily growing populations in countries such as the United States of America, there is an ensuing increase in population density near shorelines and major cities. This inherently limits the locations that can be identified for building onshore wind farms, and illuminates environmental concerns; birds, bats, noise pollution,

aesthetics, etc. Offshore wind farms, however, provide for a reduction in the potential side effects, and increase the feasible locations for future wind farms to be built. Placing the wind farms offshore also allows for larger turbines to be built that will utilize the higher wind speeds off the coast for a larger capacity of energy production. In 2008, the U.S. Offshore Wind Collaborative (USOWC) was launched, drawing representatives from many agencies and organizations [89]. This collaborative effort has the potential to accelerate growth in the offshore wind farm industry in the United States of America.

Offshore wind farms require more protection and support based on the depth of the water they inhabit and the harshness of conditions at sea. Current methods of supporting their foundations are extremely expensive, having a cost that increases proportionally with the depth of the water. There are also environmental impacts to consider. These farms have the potential to interfere with migration patterns of birds, and their foundations may also act as artificial reefs, increasing fish populations, which would likely increase the bird population in the area. Moreover, such wind farms may interfere with shipping or flight patterns, where corrective measures would be required to remove or reduce these threats. Desholm and Kahlert [21] perform a study on the ability of ducks and geese to identify and avoid wind turbines on an offshore wind farm that resided within their natural migratory pattern. Their study was conducted in the western part of the Baltic Sea near southern Denmark, and the results showed that less than 1% of the ducks and geese continued to migrate close enough to the turbines for there to be a risk of collision. Therefore, Desholm and Kahlert [21] show that the avian collision risk is relatively low, but should be considered nonetheless.

Among the environmental and social concerns relevant to offshore wind farms, economic concerns such as the specifically high operational and maintenance costs must be studied. The so-called wake effect influence, creates an impact to the short-term performance and the long-term costs of renewable energy wind farms that requires

prudence in the optimization of wind farm layouts. Renkema [69] considers the wake effect in renewable energy wind farms. This effect is created behind the wind turbine, resulting in a deceleration of wind speed and an increase in turbulence within that new airflow. Onshore wind farms are claimed to be unsuitable with any confidence to provide validation datasets for the wake effect, while this is not true for offshore wind farms [32]. This is mainly due to the lesser variation of wind speeds offshore compared to onshore. The decrease in wind speed from the wake effect reduces the performance of the downstream turbines, while the increased turbulence will reduce the lifetime of the downstream turbines. As a general rule, separating the turbines by a minimum of 10 rotor diameters will reduce the wake effect to a negligible value [69]. With initial construction costs being one of the paramount considerations in implementing offshore wind farms, maintaining a balance between the turbine spacing and the performance losses caused by the wake effect is of particular importance.

As of December 2009, the installed capacity of wind power, currently including only onshore wind farms, in the U.S. had grown to nearly 35,000 MW, which would sufficiently power 9.7 million homes. Utilizing this capacity alone reduces the nation's production of CO_2 an estimated 62 million tons, which can be converted to roughly 10.5 million cars being removed from the roadways [98]. The potential energy production of one wind turbine is approximately 1.5 to 3 MW of power [12]. Danielson [20] states that U.S. wind power capacity grew to 50,000 MW by August 2012, which could power an estimated 12 million homes each year. Danielson [20] also discloses that the U.S. wind industry received $14 billion in new investments for U.S. electric capacity additions in 2011. This shows the considerable growth potential of the wind industry in the United States. Currently there are no operational offshore wind farms along the 12,000 miles of coastline in the United States. However, the Cape Wind project, which is the first offshore wind farm to be permitted for construction off the coastline of the United States, is scheduled to begin construction in 2013 [13]. Cape

Wind will provide 130 wind turbines, producing approximately 420 MW of power using renewable wind energy. This will replace an estimated 113 million gallons of oil each year in relative nonrenewable energy consumption. With this new addition to the United States' energy independence, the research into more cost effective and efficient wind farm layouts through optimization becomes exceedingly essential to the economic growth of renewable energy production.

Optimization in wind farm planning is a balance between the maximum performance and minimum cost in a wind farm layout. The wake effect plays an important role in wind farm planning. We present a mathematical model that would minimize the cost of wind energy by examining the trade-off between the advantages of packing the turbines close together and the loss generated by wake effects.

1.3 Causal Inference with Simultaneous Denoising and Feature Selection

Regression is a statistical learning technique that develops a mathematical function that fits the data. Regression can be used for hypothesis testing, forecasting, inference, and modeling of relationships. Regression analysis is utilized in various circumstances and its significance is shown through a wide variety of studies. In its basic form, the goal of regression is to minimize a loss function that is proportional to some form of distance between data instances and the regression function. However, there are ill-posed cases where inevitable *outliers* affect the regression function in an undesired way. Our goal in this study is to introduce a novel approach that can *detect* outliers and *disregard* their contribution to the loss function. Using this approach, we are able to draw causal relations and identify relevant features in cases where outliers are most abundant such as and multiple-instance and time-series data. In the framework we consider, data consists of sets of instances that are correlated in

a way and the number of outliers in each set is bounded above.

One application that fits our framework is *learning from images* (e.g., image annotation) through segmentation. Each picture consists of a set of segments and has an underlying *response* (e.g., tumor grade for an MR image). An object to be detected that is relevant to the outcome typically does not appear in the whole image but in some segments. The goal is to identify segment(s) that is/are correlated with the response, disregarding the rest of the segments from the same picture. Another application is *molecular activity monitoring*, where each molecule has one *response* that measures it's effectiveness on a certain activity/target. However, molecules are found in different conformations and the desired underlying effect is highlighted by only certain conformations. Likewise, in *drug activity prediction*, the aim is to find the ingredients that are responsible for the desired effect and disregard the rest. *Mining time-series data* is another area that can benefit from our framework.

One common problem with time-series data is that it rarely behaves ideally. It would be surprising to see no deviations from expected time points for certain activities (e.g., seizure time on EEG data, recession on stock market data) or finding a data set with no outliers. Time warping methods can solve the former problem to a certain extent but our approach targets both of these issues. Instead of handling single readings, multiple readings over a time window is to be considered. Readings in a small neighborhood are considered as a candidate for the underlying activity (response) during this timeframe and the remaining instances are to be disregarded. It should be noted that, information lost by disregarding some instances is expected to be minimal if the neighborhood is defined in a way that conforms to the nature of the data. The frequency of data may not be uniform for all features and/or some features may be available as averages over a timeframe. In these cases, the neighborhood can be defined as the greatest common divisor of the frequencies and responsible instances are to be detected that utilize the most suitable combination of average attributes

as well as instantaneous readings. For example, suppose we have daily readings for weather temperature together with weekly number of seizures for an epilepsy patient. Note that, features in $\bar{\mathbf{F}}$ take the same value for all data instances in the same week. The response may be available daily or weekly. We propose selecting *one daily reading from each week* (disregarding the remaining readings of the week), which defines underlying response through features in both $\mathbf{F}$ and $\bar{\mathbf{F}}$. The idea is to eliminate the effect of sudden changes in $\mathbf{F}$ that cannot be supported by $\bar{\mathbf{F}}$ as we would expect in a setting where features are available in a uniform frequency.

We use natural gas price data for computational results purposes. We want to analyze historical information on the variables that potentially have a high impact on the supply and demand for natural gas, as well as the price ($/MMBTU[1] at Henry Hub) since this model for natural gas price (Henry Hub price) is extremely useful because of its potential economical benefits for industries. Natural gas is the best fossil fuel source available to reduce greenhouse gas emissions. It emits 45% less CO_2 than coal and 27% less CO_2 than oil. Despite a 70% increase in number of houses using natural gas since 1970, greenhouse gas emissions have decreased 40% per household. Therefore, America's natural gas customers are helping the environment on carbon reduction and their leading in energy efficiency. Natural gas is an important ingredient for production of many other sustainable energy sources. It is used to manufacture lightweight steel for fuel-efficient cars and trucks, to produce hydrogen for fuel cells, as a component of windmill blades for wind energy and to grow the corn needed for ethanol. Natural gas is also a backup fuel source for intermittent solar and wind energy. We use our optimization algorithm to simultaneously remove noise and select actual features affecting natural gas price.

[1]MMBTU: One million British thermal units. One British thermal unit is approximately 1055 joules.

1.4 Chapter Organization

This dissertation is divided into five chapters. We introduce the brief background and motivation as well as the problem definition in Chapter 1. In Chapter 2 an extensive literature review of robust support vector machines and multiple instance learning are presented. We first formulate the problem using different linear and nonlinear integer programming and constraint programming approaches. Next, we develop a three-phase heuristic. The performance of our approach in terms of time and generalization is shown in the computational result part of this chapter. Offshore wind farm layout optimization is studied in Chapter 3. Two different formulations are proposed that consider wake effect model in locating the wind turbines. We show the cost per energy result for two available public data set and illustrate the best layout for each of them. In Chapter 4 we first present a literature review for regression and feature selection approaches. Next, we explain our linear and nonlinear regression with denoising formulations and develop an algorithm to do feature selection while denoising the data. Finally, natural gas data have been used to show the performance of our approach. In Chapter 5, we summarize our research and then, the future work of this research is explained.

Chapter 2 Robust Support Vector Machines for Multiple Instance Learning

2.1 Introduction

As explained, multiple instances learning can be used for image retrieval of energy related images. To solve this classification problem for MIL data, a number of different approaches have been proposed. Employed methods include diverse density, decision trees, nearest neighbor algorithm, and support vector machines. In this dissertation, we propose a robust approach for MIL based on hard margin Support Vector Machine (SVM) formulations[1]. Cross validation results show that our approach provides more accurate predictions than a traditional SVM approach to MIL. In general, the term *robustness* implies a non-drastic change in performance under different settings such as noisy environment or worst case scenario depending on the context. In the context of classification, we use *robustness* to indicate minimal influence of *outliers* on the classifier, thus better generalization performance.

This chapter is organized as follows: In Section 2.2, we provide basics of SVM with different loss functions, MIL, and a brief literature survey. Section 2.3 defines the problem and presents exact integer programing and constraint programming formulations. In Section 2.4, we propose a three-phase heuristic to be used for larger problems for both linear and nonlinear classification. Section 2.5 presents the optimality performance of our heuristic and cross validation results for the proposed approach on linear and nonlinear classification of publicly available data sets. In order to show the hard margin loss is of the essence for robustness, we also demonstrate cross validation results for linear classification using hinge loss, ramp loss, and hard

[1] An earlier version of this chapter is published in [66]

margin loss on randomly generated data sets. We used offshore wind farm site location data to show the implementation of our method. We provide brief summary in Section 2.6.

2.2 Background

2.2.1 Support Vector Machines

SVMs are supervised machine learning methods that are originally used to classify pattern vectors which belong to two linearly separable sets from two different classes [90]. The classification is achieved by a hyperplane that maximizes the distance between the convex hulls of both classes. Although extensions are proposed for regression and multi-class classification, SVMs are particularly useful for binary (2-class) classification due to strong fundamentals from the statistical learning theory, implementation advantages (e.g., sparsity), and generalization performance. When misclassified instances are penalized in the linear form, SVM classifiers are proven to be universally consistent [79]. A classifier is *consistent* if the probability of misclassification (in expectation) converges to a Bayes' optimal rule when the number of data instances increase. A classifier is *universally consistent* if it is consistent for all distributions of data. SVMs can also perform nonlinear classification utilizing separating curves by implicitly embedding original data in a nonlinear space using *kernel functions*. SVMs have a wide range of applications including pattern recognition [11], text categorization [40], biomedicine [9, 47, 63], brain-computer interface [58, 47], and financial applications [87, 35].

In a typical *binary classification* problem, class $\mathbf{S}^+$ and $\mathbf{S}^-$ are composed of pattern vectors $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$. If $\mathbf{x}_i \in \mathbf{S}^+$, it is given the label $y_i = 1$; if $\mathbf{x}_i \in \mathbf{S}^-$, then it is given the label $y_i = -1$. The ultimate goal is to determine which class a new pattern vector $\mathbf{x}_i \notin \{\mathbf{S}^+ \cup \mathbf{S}^-\}$ belongs to. SVM classifiers solve this problem

by finding a hyperplane $(\mathbf{w}, b)$ that separates instances in classes $\mathbf{S}^+$ and $\mathbf{S}^-$ with the maximum interclass margin. The original *hinge loss* 2-class SVM problem is as follows:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \in I \tag{2.1a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \qquad \forall i \in I \tag{2.1b}$$

$$\xi_i \geq 0 \qquad \forall i \in I. \tag{2.1c}$$

In this formulation, $\mathbf{w}$ is the normal vector and b is the offset parameter for the separating hyperplane. ξ_i are slack variables for misclassified pattern vectors and I is a set of all instances. The goal is to maximize the interclass margin [2] and minimize misclassification. The role of scalar C in the objective function is to control the trade-off between margin violation and regularization. It should be noted that parameter C might differ for positive and negative class (e.g., C_1 and C_2) to cover *unbalanced* classification problems.

Lagrangian dual formulation for (2.1) leads to an optimization problem where input vectors only appear in the form of dot products and a suitable kernel function can be introduced for nonlinear classification [19]. This dual problem is a concave maximization problem, which can be solved efficiently. The dual for hinge loss formulation in (2.1) is given as

$$\max_{\boldsymbol{\alpha},b} \quad \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{2.2a}$$

$$\text{subject to} \quad \sum_i y_i \alpha_i = 0 \tag{2.2b}$$

$$0 \leq \alpha_i \leq C \qquad \forall i \in I. \tag{2.2c}$$

[2] Maximizing interclass margin is identical to minimizing $\|\mathbf{w}\|$ when functional distance $\langle \mathbf{w}, \mathbf{x}_i \rangle + b$ is bounded as in (2.1b). See [90] for details.

Using a hinge loss function for ξ_i as in (2.1a) or a quadratic loss function results in an increased sensitivity to outliers due to penalization of continuous measure of misclassification [8, 86, 101]. Different loss functions are proposed in the literature to model a better representation of the outliers that leads to more robust classifiers. These functions ensure that the distance from the hyperplane has a limited (if any) effect on the quality of the solution for misclassified instances. For instance, *hard margin loss* considers the number of misclassifications instead of their distances to the hyperplane [8]. Minimizing the number of misclassified points is proven to be $\mathcal{NP}$-hard [14]. Orsenigo and Vercellis [64] use a similar approach called discrete SVM (DSVM), and propose a heuristic algorithm to generate local optimum decision trees. Recently, Brooks [8] formulate the hard margin loss formulation using a set of binary variables v_i, which are equal to one if the instance is misclassified,

$$\min_{\mathbf{w},b,\boldsymbol{v}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i v_i \tag{2.3a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1, \text{if } v_i = 0 \qquad \forall i \in I \tag{2.3b}$$

$$v_i \in \{0,1\} \qquad \forall i \in I. \tag{2.3c}$$

Constraints (2.3b) can be linearized using a sufficiently large constant M as follows:

$$\min_{\mathbf{w},b,\boldsymbol{v}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i v_i \tag{2.4a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_i \qquad \forall i \in I \tag{2.4b}$$

$$v_i \in \{0,1\} \qquad \forall i \in I. \tag{2.4c}$$

In SVM classifiers, functional distance (i.e., $\langle \mathbf{w}, \mathbf{x}_i\rangle + b$) is expected to be equal to 1 (-1) for correctly classified *positive (negative) labeled instances that provide support*. Therefore, a positive and a negative labeled instance can be on the desired sides of the hyperplane yet incur misclassification penalties when functional distances are in $(0,1)$ and $(-1,0)$, respectively. In order to smooth out this effect, an approach is to

penalize misclassified instances with a functional distance in $(-1, 1)$ based on their distance and incur a fixed penalty for those out of $(-1, 1)$ range [8, 56]. This approach is called *ramp loss* or *robust hinge loss*, which can be formulated as

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{v}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_i \xi_i + 2\sum_i v_i) \tag{2.5a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \text{ if } v_i = 0 \qquad \forall i \in I \tag{2.5b}$$

$$v_i \in \{0, 1\} \qquad \forall i \in I \tag{2.5c}$$

$$0 \leq \xi_i \leq 2 \qquad \forall i \in I, \tag{2.5d}$$

where the conditional constraint (2.5b) can be linearized using M as follows:

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{v}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_i \xi_i + 2\sum_i v_i) \tag{2.6a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i - Mv_i \qquad \forall i \in I \tag{2.6b}$$

$$v_i \in \{0, 1\} \qquad \forall i \in I \tag{2.6c}$$

$$0 \leq \xi_i \leq 2 \qquad \forall i \in I. \tag{2.6d}$$

Shen et al. [78] use optimization with ramp loss but the solution method does not guarantee global optimality. Xu et al. [101] solve the non-convex optimization problem using semi-definite programming techniques but state that the procedure works inefficiently. Wang et al. [95] propose a concave-convex procedure (CCCP) to transform the associated non-convex optimization problem into a convex problem and use Newton optimization technique in the primal space. Next, we focus on MIL and present methods that are employed highlighting a set of SVM studies.

2.2.2 Multiple Instance Learning

The MIL setting is introduced by Dietterich et al. [22] for the task of drug activity prediction and design. Same setting has also been studied for applications such as identification of proteins [85], content based image retrieval [106], object detection [94], prediction of failures in hard drives [60] and text categorization [3]. In contrast to a typical classification setting where instance labels are known with certainty, MIL deals with uncertainty in labels. In multiple instance binary classification, a positive bag label shows that there is at least one actual positive instance in the bag which is a *witness* for the label. On the other hand, all instances in a negative bag must belong to the negative class so there is no uncertainty on negative labeled bags.

Several methods have been applied to solve MIL problems, from expectation maximization methods with diverse density (EM-DD) [15, 105], to deterministic annealing [29], to extensions of k-NN, citation k-NN, and diverse density methods [23], to kernel based SVM methods [3].

SVM methods have first been employed by Andrews et al. [3] for MIL. In this study, integer variables are used to indicate witness status of points in positive bags. Witness point has to be placed on the positive side of the decision boundary, otherwise a penalty is incurred. Selecting each of these representations leads to a heuristic for solving the resulting mixed-integer program approximately. In contrast, Mangasarian and Wild [52] introduce continuous variables to represent the convex combination of each positive bag, which must be placed on the positive side of the separating plane. This representation leads to an optimization problem that contains both linear and bilinear constraints, which is solved to a local optimum solution through a linear programming algorithm. An integer programming formulation that penalizes negative labeled instances without a bag notion is proposed in [44]. The setting leads to a maximum margin hyperplane between a selection of instances from positive bags and

all instances from negative bags. This problem is proven to be $\mathcal{NP}$-hard and a branch and bound algorithm is proposed.

Next, we introduce our robust classification approach through different hard margin loss formulations for MIL.

2.3 Mathematical Modeling

Despite the large number of approaches for MIL, to the best of our knowledge, our study is the first one that utilizes a robust SVM classifier for MIL. Instead of a continuous measure for misclassification, we use a hard margin loss formulation and minimize the number of misclassified instances to overcome the aforementioned outlier sensitivity issue.

The data consists of pattern vectors (instances) $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$ and bags $j = 1, \ldots, m$. Each data instance belongs to one bag. Bags are labeled positive or negative and sets of positive and negative bags are represented as $J^+ = \{j : y_j = 1\}$ and $J^- = \{j : y_j = -1\}$, respectively. Note that, labels y_j are associated with bags, rather than instances. Next, we introduce instances in positive and negative bags as $I^+ = \{i : i \in I_j \wedge j \in J^+\}$, $I^- = \{i : i \in I_j \wedge j \in J^-\}$, respectively. The goal in our robust SVM model is to maximize the interclass margin where a fixed penalty (independent from the distance) is incurred for a bag if

- the bag is positive labeled and all instances in the bag are misclassified (on the negative side),
- the bag is negative labeled and at least one instance in the bag is misclassified (on the positive side).

Here we present three integer programming and two constraint programming formulations for the described model.

2.3.1 Integer Programming Formulations

In order to use hard margin loss for multiple instance data, we define a set of variables η_i to indicate actual positive instances from each positive bag. η_i is one when we select positive instance i (as witness) and zero otherwise. We consider one selected instance from each positive bag as the witness of all instances in that bag. In order to incorporate the effect of misclassifying a bag in the objective function, we introduce two sets of variables v_j^+, v_j^- that indicate misclassification of positive and negative bags, respectively. A positive bag is misclassified ($v_j^+ = 1$) if all the instances in that positive bag is misclassified ($v_i = 1 \; \forall i \in I_j, j \in J^+$). A negative bag is misclassified ($v_j^- = 1$) if at least one instance in that bag is misclassified ($\exists i \in I_j, j \in J^- | v_i = 1$). Therefore, the multiple instance hard margin SVM (MIHMSVM) can be formulated as follows:

$$
\begin{aligned}
\textbf{MIHMSVM} \quad & \min_{\mathbf{w},b,\boldsymbol{\eta},\boldsymbol{v},\boldsymbol{v}^-} && \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{i\in I^+} v_i && && \text{(2.7a)}\\
& \text{subject to} && -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_i && \forall i \in I^- && \text{(2.7b)}\\
& && \langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - Mv_i - M(1-\eta_i) && \forall i \in I^+ && \text{(2.7c)}\\
& && \sum_{i\in I_j} \eta_i = 1 && \forall j \in J^+ && \text{(2.7d)}\\
& && v_i \leq v_j^- && \forall j \in J^-, i \in I_j && \text{(2.7e)}\\
& && 0 \leq v_j^- \leq 1 && \forall j \in J^- && \text{(2.7f)}\\
& && v_i \in \{0,1\} && \forall i \in I^+ \cup I^- && \text{(2.7g)}\\
& && \eta_i \in \{0,1\} && \forall i \in I^+. && \text{(2.7h)}
\end{aligned}
$$

In this formulation, (2.7c) is always satisfied for all positive instances that are not witnesses (i.e., $\eta_i = 0$), which sets $v_i = 0$ due to nature of the objective function. Therefore, only the witness of a positive bag with $\eta_i = 1$ might deteriorate the objective function. This ensures that a positive bag does not incur any penalty if at

least one instance is correctly classified. On the other hand, v_i values for negative instances are calculated as in a typical classification problem. However, (2.7e) ensures that the maximum of these values are penalized in the objective function and a negative bag does not incur a penalty if all instances are correctly classified. It should be noted that MIHMSVM is $\mathcal{NP}$-hard since a special case with a single instance in each bag is proven to be $\mathcal{NP}$-hard [52].

This formulation utilizes $2|I^+| + |I^-|$ binary variables and $|J^-|$ continuous variables. Instead of using constraints (2.7e), we can use the binaries inside separation constraints directly. This will not only reduce the number of binary variables, but eliminate the need for continuous variables as well. We obtain a simpler formulation with $2|I^+| + |J^-|$ binary variables as follows:

$$\textbf{IP1} \quad \min_{\mathbf{w},b,\boldsymbol{\eta},\boldsymbol{v},\boldsymbol{v}^-} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{i\in I^+} v_i \tag{2.8a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_j^- \qquad \forall j \in J^-, i \in I_j \tag{2.8b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - Mv_i - M(1-\eta_i) \qquad \forall i \in I^+ \tag{2.8c}$$

$$\sum_{i\in I_j} \eta_i = 1 \qquad \forall j \in J^+ \tag{2.8d}$$

$$v_i, \eta_i \in \{0,1\} \qquad \forall i \in I^+ \tag{2.8e}$$

$$v_j^- \in \{0,1\} \qquad \forall j \in J^-. \tag{2.8f}$$

Next formulation, influenced by Mangasarian and Wild [52], considers the fact that it is enough to select the instances with minimum misclassification from positive bags. Therefore, we utilize variables v_j^+, for positive bags that shows the minimum misclassification associated with that bag. By penalizing this variable in the objective function, we obtain

$$\min_{\mathbf{w},b,\boldsymbol{\eta},\boldsymbol{v},\boldsymbol{v}^+,\boldsymbol{v}^-} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{j\in J^+} v_j^+ \tag{2.9a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_j^- \qquad \forall j \in J^-, i \in I_j \tag{2.9b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - Mv_i \qquad \forall i \in I^+ \tag{2.9c}$$

$$v_j^+ = \sum_{i\in I_j} \eta_i v_i \qquad \forall j \in J^+ \tag{2.9d}$$

$$\sum_{i\in I_j} \eta_i = 1 \qquad \forall j \in J^+ \tag{2.9e}$$

$$v_j^+ \in \{0,1\} \qquad \forall j \in J^+ \tag{2.9f}$$

$$v_j^- \in \{0,1\} \qquad \forall j \in J^- \tag{2.9g}$$

$$v_i, \eta_i \in \{0,1\} \qquad \forall i \in I^+. \tag{2.9h}$$

In order to linearize (2.9d), we introduce new variables $\hat{z}_i$ that should be equal to $\eta_i v_i$. We relax the integrality of η_i and v_i^+ and come up with the following formulation with $|I^+| + |J^-|$ binary variables:

$$\textbf{IP2} \quad \min_{\mathbf{w},b,\boldsymbol{\eta},\boldsymbol{v},\boldsymbol{v}^+,\boldsymbol{v}^-,\hat{\boldsymbol{z}}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{j\in J^+} v_j^+ \tag{2.10a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_j^- \qquad \forall j \in J^-, i \in I_j \tag{2.10b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - Mv_i \qquad \forall i \in I^+ \tag{2.10c}$$

$$v_j^+ = \sum_{i\in I_j} \hat{z}_i \qquad \forall j \in J^+ \tag{2.10d}$$

$$\hat{z}_i \geq -1 + \eta_i + v_i \qquad \forall i \in I^+ \tag{2.10e}$$

$$\hat{z}_i \leq v_i \qquad \forall i \in I^+ \tag{2.10f}$$

$$\hat{z}_i \leq \eta_i \qquad \forall i \in I^+ \tag{2.10g}$$

$$\sum_{i\in I_j} \eta_i = 1 \qquad \forall j \in J^+ \tag{2.10h}$$

$$0 \leq v_j^+ \leq 1 \qquad \forall j \in J^+ \tag{2.10i}$$

$$0 \leq \hat{z}_i \leq 1 \qquad \forall i \in I^+ \tag{2.10j}$$

$$0 \leq \eta_i \leq 1 \qquad \forall i \in I^+ \tag{2.10k}$$

$$v_j^- \in \{0,1\} \qquad \forall j \in J^- \tag{2.10l}$$

$$v_i \in \{0,1\} \qquad \forall i \in I^+. \tag{2.10m}$$

It should be noted that constraints (2.10f) and (2.10g) are redundant since the summation of $\hat{z}_i$ is to be minimized.

Next, we obtain a novel formulation using the number of instances in positive bags to identify positive bag witnesses. Our experience with the following formulation is that it is far superior compared to **IP1** and **IP2**. We use the fact that, a positive bag is misclassified if all instances in that bag are misclassified, i.e., $\sum_{i\in I_j} v_i = |I_j|$. We also relax the integrality of v_i^+ and obtain a formulation with $|I^+| + |J^-|$ binary variables:

$$\textbf{IP3} \quad \min_{\mathbf{w},b,\boldsymbol{v},\boldsymbol{v}^+,\boldsymbol{v}^-} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{j\in J^+} v_j^+ \tag{2.11a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_j^- \qquad \forall j \in J^-, i \in I_j \tag{2.11b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - Mv_i \qquad \forall i \in I^+ \tag{2.11c}$$

$$v_j^+ \geq \sum_{i\in I_j} v_i - |I_j| + 1 \qquad \forall j \in J^+ \tag{2.11d}$$

$$0 \leq v_j^+ \leq 1 \qquad \forall j \in J^+ \tag{2.11e}$$

$$v_j^- \in \{0,1\} \qquad \forall j \in J^- \tag{2.11f}$$

$$v_i \in \{0,1\} \qquad \forall i \in I^+. \tag{2.11g}$$

Suppose j' is a positive bag with $|I_{j'}|$ instances. When all of the instances in the bag are misclassified (i.e., $v_i = 1, \forall i \in I_{j'}$) then $\sum_{i\in I_{j'}} v_i = |I_{j'}|$ and $v_i^+ = 1$ is forced. Otherwise, $\sum_{i\in I_{j'}} v_i \leq |I_{j'}| + 1$ and v_i^+ will be free and set to 0 due to the objective function.

Next, we present two constraint programming formulations for benchmarking purposes. In contrast to integer programming approaches, constraint programming prioritize exploiting special functions and finding a feasible solution during the computational procedure.

2.3.2 Constraint Programming Formulations

In order to evaluate the performance of IP formulations and take advantage of the special structure of the problem, we introduce two constraint programming formulations. IBM ILOG CPLEX CP Optimizer [37] is employed that utilize robust constraint propagation and search algorithms.

Our first constraint programming formulation is as follows:

$$\textbf{CP1} \quad \min_{\mathbf{w},b,\boldsymbol{v}^+,\boldsymbol{v}^-} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{j\in J^+} v_j^+ \tag{2.12a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 \quad \vee \quad v_j^- \geq 1 \qquad \forall j \in J^-, i \in I_j \tag{2.12b}$$

$$\bigvee_{i\in I_j} \langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 \quad \vee \quad v_j^+ \geq 1 \qquad \forall j \in J^+ \tag{2.12c}$$

$$0 \leq v_j^- \leq 1 \qquad \forall j \in J^- \tag{2.12d}$$

$$0 \leq v_j^+ \leq 1 \qquad \forall j \in J^+. \tag{2.12e}$$

In **CP1**, (2.12b) is defined for all negative labeled instances and ensures that each negative labeled instance is correctly classified OR its corresponding bag is misclassified (i.e., $v_j^- = 1$). On the other hand, (2.12c) is defined for all positive bags and forces either one of the instances in the bag to be correctly classified OR the bag is misclassified (i.e., $v_j^+ = 1$).

Next, we propose a hybrid approach using constraint programming with the constraint set from a fast IP implementation, **IP3**. The formulation is as follows:

$$\textbf{CP2} \quad \min_{\mathbf{w},b,\boldsymbol{v},\boldsymbol{v}^+,\boldsymbol{v}^-} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{j\in J^+} v_j^+ \tag{2.13a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 \quad \vee \quad v_j^- \geq 1 \qquad \forall j \in J^-, i \in I_j \tag{2.13b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 \quad \vee \quad v_i \geq 1 \qquad \forall i \in I^+ \tag{2.13c}$$

$$v_j^+ \geq \sum_{i\in I_j} v_i - |I_j| + 1 \qquad \forall j \in J^+ \tag{2.13d}$$

$$0 \leq v_j^- \leq 1 \qquad \forall j \in J^- \tag{2.13e}$$

$$0 \leq v_j^+ \leq 1 \qquad \forall j \in J^+ \tag{2.13f}$$

$$0 \leq v_i \leq 1 \qquad \forall i \in I^+. \tag{2.13g}$$

In **CP2**, constraints on bag misclassification are partially adapted from **CP1** and

IP3. Next, we present the nonlinear hard margin loss formulation for MIL.

2.3.3 Nonlinear Classification

By making the substitution $\mathbf{w} = \sum_{i=1}^{n} y_i x_i \alpha_i$ with nonnegative α_i variables for $i = 1, \ldots, n$ in (2.7), we obtain the following nonlinear classification formulation for multiple instance hard margin SVM:

$$
\begin{aligned}
\textbf{NLMIHMSVM} \quad \min_{\boldsymbol{\alpha}, b, \boldsymbol{\eta}, \boldsymbol{v}, \boldsymbol{v}^-} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \alpha_i \alpha_j + C \sum_{j \in J^-} v_j^- + C \sum_{i \in I^+} v_i && && \text{(2.14a)} \\
\text{subject to} \quad & -\sum_{j=1}^{n} y_j \langle \mathbf{x}_j, \mathbf{x}_i \rangle \alpha_j - b \geq 1 - M v_i && \forall i \in I^- && \text{(2.14b)} \\
& \sum_{j=1}^{n} y_j \langle \mathbf{x}_j, \mathbf{x}_i \rangle \alpha_j + b \geq 1 - M v_i - M(1 - \eta_i) && \forall i \in I^+ && \text{(2.14c)} \\
& \alpha_i \geq 0 && \forall i \in I^+ \cup I^- && \text{(2.14d)} \\
& \alpha_i \leq M \eta_i && \forall i \in I^+ \cup I^- && \text{(2.14e)} \\
& \sum_{i \in I_j} \eta_i = 1 && \forall j \in J^+ && \text{(2.14f)} \\
& v_i \leq v_j^- && \forall j \in J^-, i \in I_j && \text{(2.14g)} \\
& v_i \in \{0, 1\} && \forall i \in I^+ \cup I^- && \text{(2.14h)} \\
& 0 \leq v_j^- \leq 1 && \forall j \in J^- && \text{(2.14i)} \\
& \eta_i \in \{0, 1\} && \forall i \in I^+. && \text{(2.14j)}
\end{aligned}
$$

The use of (2.14) is that the original data can be embedded in a nonlinear space by replacing the dot products with a suitable kernel function $\mathbf{K}$ in (2.14a), (2.14b), and (2.14c). It should be noted that (2.14e) ensures instances that are not selected do not play a role on the hyperplane. Therefore, for a given set of $\boldsymbol{\eta}$ values, the formulation reduces to the hard margin loss formulation in [8].

Note that, both linear and nonlinear formulations presented in this section can

utilize different penalty terms to solve unbalanced classification problems. Next, we present formulations for different loss functions for the multiple instance classification problem.

2.3.4 Multiple Instance Classification with Hinge and Ramp Loss

In this section, we develop formulations for multiple instance hinge loss support vector machines and multiple instance ramp loss support vector machines for benchmarking purposes.

In order to incorporate bags in the objective function of hinge loss SVM, i.e., formulation (2.1), two sets of new variables ξ_j^+, ξ_j^- are introduced that incorporate the positive and negative bag misclassification, respectively. ξ_j^+ should be equal to minimum ξ_i in each positive bag to select the actual positive of that bag. For negative bags, ξ_j^- should be greater than or equal to each instance's ξ_i in that bag. Therefore, the problem can be formulated as

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^+,\boldsymbol{\xi}^-,\boldsymbol{\eta}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\Big(\sum_{j\in J^-}\xi_j^- + \sum_{j\in J^+}\xi_j^+\Big) \tag{2.15a}$$

$$\text{subject to} \quad -\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) \geq 1 - \xi_i \qquad \forall i \in I^- \tag{2.15b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i \qquad \forall i \in I^+ \tag{2.15c}$$

$$\xi_j^+ = \sum_{i\in I_j}\eta_i\xi_i \qquad \forall j \in J^+ \tag{2.15d}$$

$$\sum_{i\in I_j}\eta_i = 1 \qquad \forall j \in J^+ \tag{2.15e}$$

$$\xi_i \leq \xi_j^- \qquad \forall j \in J^-, i \in I_j \tag{2.15f}$$

$$\eta_i \in \{0,1\} \qquad \forall\, i \in I^+ \tag{2.15g}$$

$$\xi_i \geq 0 \qquad \forall i \in I^+ \cup I^-, \tag{2.15h}$$

which can be linearized as

$$
\begin{aligned}
\textbf{MIHLSVM} \quad \min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^+,\boldsymbol{\xi}^-,\boldsymbol{\eta},\boldsymbol{z}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_{j\in J^-}\xi_j^- + \sum_{j\in J^+}\xi_j^+) && && \text{(2.16a)}\\
\text{subject to} \quad & -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i && \forall i \in I^- && \text{(2.16b)}\\
& \langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i && \forall i \in I^+ && \text{(2.16c)}\\
& \xi_j^+ = \sum_{i\in I_j} z_i && \forall j \in J^+ && \text{(2.16d)}\\
& z_i \geq \xi_i - M(1-\eta_i) && \forall i \in I^+ && \text{(2.16e)}\\
& z_i \leq \xi_i && \forall i \in I^+ && \text{(2.16f)}\\
& z_i \leq M\eta_i && \forall i \in I^+ && \text{(2.16g)}\\
& \sum_{i\in I_j}\eta_i = 1 && \forall j \in J^+ && \text{(2.16h)}\\
& \xi_i \leq \xi_j^- && \forall j \in J^-, i \in I_j && \text{(2.16i)}\\
& \eta_i \in \{0,1\} && \forall\, i \in I^+ && \text{(2.16j)}\\
& z_i \geq 0 && \forall i \in I^+ && \text{(2.16k)}\\
& \xi_i \geq 0 && \forall i \in I^+ \cup I^-. && \text{(2.16l)}
\end{aligned}
$$

Next, we formulate ramp loss for MIL. Similar to the previous formulations, variables $\xi_j^+, \xi_j^-, v_j^+, v_j^-$ are defined to incorporate the misclassification of positive and negative bags with the ramp loss definition discussed in Section (2.2). The resulting formulation for ramp loss SVM for MIL data is

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^+,\boldsymbol{\xi}^-,\boldsymbol{v}^+,\boldsymbol{v}^-,\boldsymbol{v},\boldsymbol{\eta}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_{j\in J^-}\xi_j^- + \sum_{j\in J^+}\xi_j^+ + 2\sum_{j\in J^-}v_j^- + 2\sum_{j\in J^+}v_j^+) \tag{2.17a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i - Mv_j^- \qquad \forall j \in J^-, i \in I_j \tag{2.17b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i - Mv_i \qquad \forall i \in I^+ \tag{2.17c}$$

$$\xi_j^+ = \sum_{i\in I_j}\eta_i\xi_i \qquad \forall j \in J^+ \tag{2.17d}$$

$$v_j^+ = \sum_{i\in I_j}\eta_i v_i \qquad \forall j \in J^+ \tag{2.17e}$$

$$\sum_{i\in I_j}\eta_i = 1 \qquad \forall j \in J^+ \tag{2.17f}$$

$$\xi_i \leq \xi_j^- \qquad \forall j \in J^-, i \in I_j \tag{2.17g}$$

$$v_i, \eta_i \in \{0,1\} \qquad \forall i \in I^+ \tag{2.17h}$$

$$v_j^+ \in \{0,1\} \qquad \forall j \in J^+ \tag{2.17i}$$

$$v_j^- \in \{0,1\} \qquad \forall j \in J^- \tag{2.17j}$$

$$0 \leq \xi_i \leq 2 \qquad \forall i \in I^+ \cup I^-, \tag{2.17k}$$

which can be linearized using two sets of variables,

$$\gamma_j^+ = \xi_j^+ + 2v_j^+ \quad \forall j \in J^+$$

$$\gamma_j^- = \xi_j^- + 2v_j^- \quad \forall j \in J^-,$$

as follows:

$$\textbf{MIRLSVM} \quad \min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\gamma}^+,\boldsymbol{\gamma}^-,\boldsymbol{v}^-,\boldsymbol{v},\boldsymbol{\eta},\boldsymbol{z}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_{j\in J^-}\gamma_j^- + \sum_{j\in J^+}\gamma_j^+ \tag{2.18a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i - Mv_j^- \quad \forall j \in J^-, i \in I_j \tag{2.18b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i - Mv_i \quad \forall i \in I^+ \tag{2.18c}$$

$$\gamma_j^+ = \sum_{i\in I_j} z_i \quad \forall j \in J^+ \tag{2.18d}$$

$$z_i \geq (\xi_i + 2v_i) - M(1-\eta_i) \quad \forall i \in I^+ \tag{2.18e}$$

$$z_i \leq (\xi_i + 2v_i) \quad \forall i \in I^+ \tag{2.18f}$$

$$z_i \leq M\eta_i \quad \forall i \in I^+ \tag{2.18g}$$

$$\sum_{i\in I_j}\eta_i = 1 \quad \forall j \in J^+ \tag{2.18h}$$

$$2v_i^- + \xi_i \leq \gamma_j^- \quad \forall j \in J^-, i \in I_j \tag{2.18i}$$

$$v_i, \eta_i \in \{0,1\} \quad \forall i \in I^+ \tag{2.18j}$$

$$v_j^- \in \{0,1\} \quad \forall j \in J^- \tag{2.18k}$$

$$0 \leq \xi_i \leq 2 \quad \forall i \in I^+ \cup I^- \tag{2.18l}$$

$$z_i \geq 0 \quad \forall i \in I^+. \tag{2.18m}$$

Next section presents a heuristic algorithm for larger problems to be solved using hard margin loss formulation which is $\mathcal{NP}$-hard and exact methods may be computationally intractable.

2.4 Three-Phase Heuristic Algorithm

In this section, we develop a three-phase heuristic for the proposed MIHMSVM model. First, we explore the details of the algorithm for linear classification and present the pseudocode. Next, we highlight the modifications needed to perform nonlinear classification.

2.4.1 Linear Classification

The idea of our algorithm is to start with a feasible hyperplane and fine tune the orientation considering MIL restrictions. Instead of starting with a random hyperplane, we take advantage of the efficiency of SVM on a typical classification problem. Therefore, the first phase of the algorithm consists of applying hinge loss SVM classifier on all instances considering their labels regardless of their bags. We use LIBSVM [34] since a fast classification of the data set is needed. The optimal separating hyperplane in this step $(\mathbf{w}_1, b_1)$ gives a rough idea on positioning of bags. Next, we select a representative for each bag. Bag representatives may be interpreted as witnesses for positive bags. Although MIL setting does not entail negative bag witnesses, the reason we select representatives for negative bags is to keep the number of positive and negative labeled instances balanced and avoid biased classifications for the next step. The choice of bag representatives is based on the maximum functional distance from the hyperplane, which is in line with margin maximization objective considering MIL setting. This approach provides furthest correctly classified (or least misclassified) instances in positive bags and closest correctly classified (or most misclassified) instances in negative bags as representatives. Next, we use hinge loss SVM classifier for selected instances from all bags. The optimal separating hyperplane of this step is $(\mathbf{w}_2, b_2)$ that supposedly gives a better representation of data. This classifier will be used to find the correctly classified negative bags (where all instances are on negative side) and positive bags (where at least one instance is on positive side) as an initial solution at the end of the first phase.

In the second phase, a hard separation problem is solved. The instance with maximum functional distance from $(\mathbf{w}_2, b_2)$ in each correctly classified positive bag constitute the positive labeled training set. On the other hand, all instances in correctly classified negative bags are included in the negative labeled training set.

Note that, a hard separation problem (i.e., formulation (2.3) where $v_i = 0, \forall i$) is polynomially solvable, and the resulting solution from phase one assures there will be no misclassification at this step. Since there are no misclassification terms for instances, an imbalance (possibly large number of negative labeled instances) does not imply a biased classifier. Let $(\mathbf{w}_3, b_3)$ be the optimal separating hyperplane at the end of this step. Next, we search for fast inclusion of misclassified bags while maintaining feasibility of the hard separation problem by fixing $(\mathbf{w}_3, b_3)$. Finally, we compute current objective function value of MIHMSVM using $\|\mathbf{w}_3\|^2$ and number of misclassified bags. This hyperplane also becomes the *current* best solution.

In the third (improvement) phase, we employ a more rigorous inclusion process. Misclassified bags are sorted in ascending order of their distance from their corresponding *support* hyperplane and considered as *candidates* to be correctly classified one by one. Distance between a positive bag and the support hyperplane is defined as the distance between closest instance and the positive support hyperplane (i.e., $\langle \mathbf{w}, \mathbf{x}_i \rangle + b = 1$). On the other hand, distance between a negative bag and the support hyperplane is defined as the distance between furthest instance and the negative support hyperplane (i.e., $\langle \mathbf{w}, \mathbf{x}_i \rangle + b = -1$). This approach is in line with our model assumptions in Section 2.3. If a positive bag is considered, instance with the smallest distance will be temporarily added to the training set. If a negative bag is selected, all instances in the bag will be temporarily added to the training set. Next, training set is examined for feasibility and if the problem is feasible, hyperplane $(\mathbf{w}_4, b_4)$ is obtained. If hard margin loss objective function is less than the current best objective, candidate bag will be added to the solution and best hyperplane is updated. The objective functions are compared based on the fact that by adding a bag, we decrease the misclassification by one in trade of a change in the norm of the hyperplane. Thus, in an iteration, if $(\|\mathbf{w}_4\|^2 - \|\mathbf{w}_{best}\|^2)/2$ is less than C, then we conclude the overall objective is reduced. The search will continue until no improvement is possible and

the final best solution is the heuristic solution for the problem. The pseudocode is presented in Algorithm 1.

Algorithm 1 Three-Phase Heuristic Algorithm (Linear Classification)
INPUT: $\mathbf{x}_1, \ldots, \mathbf{x}_n$, J^+, J^-, I^+, I^-, C
OUTPUT: $\mathbf{w}_{best}, b_{best}, Objective$

{**PHASE I**}
$P \leftarrow I^+$
$N \leftarrow I^-$
$\mathbf{w}_1, b_1 \leftarrow$ regular hinge-loss SVM hyperplane that separates P and N
Empty P and N
for all $j \in J^+$ **do**
  $P \leftarrow P \cup arg\max_{i \in I_j} \langle \mathbf{w}_1, x_i \rangle + b_1$
end for
for all $j \in J^-$ **do**
  $N \leftarrow N \cup arg\max_{i \in I_j} \langle \mathbf{w}_1, x_i \rangle + b_1$
end for
$\mathbf{w}_2, b_2 \leftarrow$ regular hinge-loss SVM hyperplane that separates P and N

{**PHASE II**}
Empty P and N
$number\ of\ misclassified\ bags \leftarrow 0$
for all $j \in J^+$ **do**
  if $\max_{i \in I_j} \langle \mathbf{w}_2, x_i \rangle + b_2 > 0$ **then**
    $P \leftarrow P \cup arg\max_{i \in I_j} \langle \mathbf{w}_2, x_i \rangle + b_2$
  else
    $number\ of\ misclassified\ bags \leftarrow number\ of\ misclassified\ bags + 1$
  end if
end for
for all $j \in J^-$ **do**
  if $\max_{i \in I_j} \langle \mathbf{w}_2, x_i \rangle + b_2 < 0$ **then**
    $N \leftarrow N \cup I_j$
  else
    $number\ of\ misclassified\ bags \leftarrow number\ of\ misclassified\ bags + 1$
  end if
end for
$\mathbf{w}_3, b_3 \leftarrow$ hard separation SVM hyperplane that separates P and N
{***Fast Inclusion***}
for all $j \in J^+$ **do**
  if $I_j \cap P = \emptyset$ AND $\max_{i \in I_j} \langle \mathbf{w}_3, x_i \rangle + b_3 > 1$ **then**
    $P \leftarrow P \cup arg\max_{i \in I_j} \langle \mathbf{w}_3, x_i \rangle + b_3$
    $number\ of\ misclassified\ bags \leftarrow number\ of\ misclassified\ bags - 1$
  end if
end for
for all $j \in J^-$ **do**
  if $I_j \cap N = \emptyset$ AND $\max_{i \in I_j} \langle \mathbf{w}_3, x_i \rangle + b_3 < -1$ **then**
    $N \leftarrow N \cup I_j$
    $number\ of\ misclassified\ bags \leftarrow number\ of\ misclassified\ bags - 1$
  end if
end for
$Objective \leftarrow \frac{1}{2}\|\mathbf{w}_3\|^2 + C \times number\ of\ misclassified\ bags$

{**PHASE III**}
$active\_set \leftarrow \emptyset$
$\mathbf{w}_{best} \leftarrow \mathbf{w}_3$
$b_{best} \leftarrow b_3$
for all $j \in (J^+ \cup J^-)$ **do**
  if $I_j \cap (P \cup N) \neq \emptyset$ **then**
    $active\_set \leftarrow active\_set \cup j$
  end if
end for
while $active\_set \neq \emptyset$ **do**
  if $\min_{j \in (active\_set \cap J^+)}[-\max_{i \in I_j}(\langle \mathbf{w}_{best}, x_i \rangle + b_{best} - 1)] < \min_{j \in (active\_set \cap J^-)}[\max_{i \in I_j}(\langle \mathbf{w}_{best}, x_i \rangle + b_{best} + 1)]$ **then**
    $candidate \leftarrow \ arg\min_{j \in (active\_set \cap J^+)}[-\max_{i \in I_j}(\langle \mathbf{w}_{best}, x_i \rangle + b_{best} - 1)]$
    $P \leftarrow P \cup arg\max_{i \in I_{candidate}}(\langle \mathbf{w}_{best}, x_i \rangle + b_{best} - 1)$
  else
    $candidate \leftarrow \ arg\min_{j \in (active\_set \cap J^-)}[\max_{i \in I_j}(\langle \mathbf{w}_{best}, x_i \rangle + b_{best} + 1)]$
    $N \leftarrow N \cup I_{candidate}$
  end if
  $active\_set \leftarrow active\_set \setminus candidate$
  if hard separation for P and N is feasible **then**
    $\mathbf{w}_4, b_4 \leftarrow$ hard separation SVM hyperplane that separates P and N
    if $\frac{1}{2}\|w_4\|^2 - \frac{1}{2}\|w_{best}\|^2 < C$ **then**
      $\mathbf{w}_{best} \leftarrow \mathbf{w}_4$
      $b_{best} \leftarrow b_4$
      $Objective \leftarrow Objective + \frac{1}{2}\|w_4\|^2 - \frac{1}{2}\|w_{best}\|^2 - C$
    else
      $P \leftarrow P \setminus I_{candidate}$
      $N \leftarrow N \setminus I_{candidate}$
    end if
  else
    $P \leftarrow P \setminus I_{candidate}$
    $N \leftarrow N \setminus I_{candidate}$
  end if
end while

2.4.2 Nonlinear Classification

Nonlinear extension of Algorithm 1 utilizes a number of modifications. In the first phase, regular hinge loss SVM is substituted with nonlinear SVM with a kernel function to obtain $(\boldsymbol{\alpha}_1, b_1)$. Next, in the construction of P and N, $\langle \mathbf{w}_1, x_i \rangle$ are substituted with $\sum_{j=1}^{n} y_j \mathbf{K}(\mathbf{x}_j, \mathbf{x}_i)\alpha_{1j}$ to calculate the distances. At the last step of the first phase, nonlinear SVM with kernel is employed again to obtain $(\boldsymbol{\alpha}_2, b_2)$. Likewise, in the second phase, $\langle \mathbf{w}_2, x_i \rangle$ are substituted with $\sum_{j=1}^{n} y_j \mathbf{K}(\mathbf{x}_j, \mathbf{x}_i)\alpha_{2j}$.

In order to obtain a nonlinear hard separation in Phase 2, we used the following formulation based on [8]:

$$\min_{\boldsymbol{\alpha},b} \quad \frac{1}{2}\sum_{i\in P\cup N}\sum_{j\in P\cup N} y_i y_j \mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)\alpha_i\alpha_j \tag{2.19a}$$

$$\text{subject to} \quad \sum_{j\in P\cup N} y_j \mathbf{K}(\mathbf{x}_j,\mathbf{x}_i)\alpha_j + b \geq 1 \qquad \forall i \in P \tag{2.19b}$$

$$-\sum_{j\in P\cup N} y_j \mathbf{K}(\mathbf{x}_j,\mathbf{x}_i)\alpha_j - b \geq 1 \qquad \forall i \in N \tag{2.19c}$$

$$\alpha_i \geq 0 \qquad \forall i \in P\cup N. \tag{2.19d}$$

Optimal solution to (2.19) provides $(\boldsymbol{\alpha}_3, b_3)$ that is used for fast inclusion. For distance calculation and in order to ensure hard separability, $\langle \mathbf{w}_3, x_i\rangle$ are substituted with $\sum_{j=1}^{n} y_j \mathbf{K}(\mathbf{x}_j,\mathbf{x}_i)\alpha_{3j}$. At the last step of Phase 2, $\|\mathbf{w}_3\|^2$ is substituted with the optimal objective function value of (2.19), i.e., $1/2\sum_{i\in P\cup N}\sum_{j\in P\cup N} y_i y_j \mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)\alpha_{3i}\alpha_{3j}$.

As expected, in the third phase, instead of working with $\mathbf{w}$, we keep considering $\boldsymbol{\alpha}$ vectors. Decision of *candidate* instance for inclusion is performed by substituting dot products $\langle \mathbf{w}_{best}, x_i\rangle$ with $\sum_{j=1}^{n} y_j \mathbf{K}(\mathbf{x}_j,\mathbf{x}_i)\alpha_{bestj}$. Hard separation with $(\mathbf{w}_4, b_4)$ is also substituted with $(\boldsymbol{\alpha}_4, b_4)$ which gets the optimal solution for formulation (2.19).

Next, we report computational performance for the proposed algorithm. We also show hard margin loss is virtually more robust and better in terms of generalization performance compared to other loss functions.

2.5 Computational Results

In this section, we first present the superior performance of hard margin loss in practice compared to ramp and hinge loss functions using randomly generated data sets. Next, we evaluate the performance of our heuristic in terms of time and proximity to the optimal solution. Then, we show the cross validation performance of the proposed heuristic on the publicly available data sets. Finally, we implement our method to wind farm site locating problem. All computations are performed on a 2.93

GHz Intel Core 2 Duo computer with 4.0 GB RAM. The algorithms are implemented in C++ and used in conjunction with MATLAB 7.11.0 (2010b) [57] environment in which the data resides.

We use MUSK1 and MUSK2 data set from UCI Machine Learning Repository [27]. MUSK1 data set consists of descriptions of 92 molecules (bags) with different shapes or conformations. Among them 47 of molecules judged by human experts are labeled as musks (positive bags) and remaining 45 molecules are labeled as non-musks (negative bags). The total number of conformations (instances) are 476 that gives an average of 5.2 conformations for each molecule (bag). MUSK2 data set consists of descriptions of 102 molecules in which 39 of molecules are labeled as musks and remaining 63 molecules are labeled as non-musks. Total number of conformations is 6,598 which gives an average of 64.7 conformations for each molecule. Each conformation in data sets is represented with a vector of 166 features extracted from surface properties.

Leave One Bag Out Cross Validation

Traditional cross validation methods (e.g., leave one out, n-fold) cannot reflect a fair assessment of multiple instance approaches due to ambiguity with actual instance labels. Therefore, we employ an extension that we refer to as *leave one bag out cross validation* (LOBOCV), which uses one bag from the original data set for validation (test data) and remaining instances as training data. After the separating hyperplane is obtained, label of the test bag is predicted and compared with its actual label. This routine is repeated until each bag in the sample is validated once and the percentage of correctly classified bags is reported.

2.5.1 Robustness of MIHMSVM

The robustness of the objectives will be discussed based on randomly generated data and the results obtained using IBM ILOG CPLEX Optimization Studio 12.2 [37]. Table 2.1 shows the cross validation results for three loss functions presented, namely hard margin loss (MIHMSVM) in (2.7), ramp loss (MIRLSVM) in (2.18), and hinge loss (MIHLSVM) in (2.16). In our computational studies, we consider a number of different C values. Small values result in a larger number of misclassified bags, which is not desired. On the other hand, values greater than 1 do not lead to a drastic decrease in the number of misclassifications (see [8]). Therefore, we set $C = 1$ for our experiments in this section. This penalty parameter also provides the best generalization performance for larger data sets, as shown in Section 2.5.3. Problem instances are generated using predetermined number of bags and features and the following pattern vector distributions:

TB1 Normal distribution: Features for instances in negative bags are normally distributed with mean 0, standard deviation 1. The mean of features for a positive bag are normally distributed with mean 1, standard deviation 5, and instances within each positive bag are offset using a normal distribution with mean 0, standard deviation 1. There are 4 instances in each positive and negative bag.

TB2 Uniform distribution: Features for instances in negative bags are uniformly distributed between -1 and 2. The mean of features for a positive bag are uniformly distributed between -2 and 4, and instances within each positive bag are offset uniformly between -1 and 1. There are 4 instances in each positive and negative bag.

TB3 Randomly selected features and bags from MUSK1 data set.

Table 2.1: Leave-one-bag-out cross validation results for randomly generated multiple instance learning problems using different loss functions.

Testbed	# of Bags	# of Features	Hard Margin Loss (MIHMSVM)	Ramp Loss (MIRLSVM)	Hinge Loss (MIHLSVM)
TB1	15	60	60.00%	60.00%	60.00%
TB2	15	60	80.00%	80.00%	80.00%
TB3	15	60	46.67%	**53.33%**	**53.33%**
TB3	15	60	80.00%	80.00%	80.00%
TB3	15	60	66.67%	66.67%	66.67%
TB1	20	80	50.00%	50.00%	50.00%
TB2	20	80	55.00%	55.00%	55.00%
TB3	20	80	**65.00%**	50.00%	50.00%
TB3	20	80	**45.00%**	40.00%	40.00%
TB3	20	80	**40.00%**	35.00%	35.00%
TB1	25	80	80.00%	80.00%	80.00%
TB2	25	80	88.00%	88.00%	88.00%
TB3	25	80	**56.00%**	36.00%	40.00%
TB3	25	80	**64.00%**	44.00%	40.00%
TB3	25	80	**56.00%**	36.00%	40.00%

The results shows hard margin loss is usually superior in practice compared to other loss functions. Loss functions would have minimal effect on classifiers for easy problems where a clean separation is possible. This can be observed in Table 2.1 when the ratio of number of instances to number of features is relatively low. In fact, for all cases created using TB1 and TB2, we observe the same accuracy for all three loss functions, which are not presented due to space considerations. This behavior changes for (i) odd distributions with outliers, (ii) when there are bags with small number of instances, and (iii) when the ratio of number of instances to number of features is higher. This directly points to MUSK1 data set with larger number of instances as can be seen in the last few rows of Table 2.1. In order to show this effect on relatively smaller instances, we generate the following instances by injecting outliers:

TB1o Normal distribution: Features for instances in negative bags are normally distributed with mean 0, standard deviation 4. The mean of features for a positive bag are normally distributed with mean 5, standard deviation 4. There are 4 instances in each positive and negative bag. One out of five negative bags are

injected one noisy instance that is normally distributed with mean ±90 and standard deviation 2.

TB2o Uniform distribution: Features for instances in negative bags are uniformly distributed between -10 and 10. The mean of features for a positive bag are uniformly distributed between -5 and 15. There are 4 instances in each positive and negative bag. One out of five negative bags are injected one noisy instance that is uniformly distributed between ±(80,100).

Table 2.2: Leave-one-bag-out cross validation results for randomly generated multiple instance learning problems with outliers using different loss functions.

Testbed	**# of Bags**	**# of Features**	**Hard Margin Loss (MIHMSVM)**	**Ramp Loss (MIRLSVM)**	**Hinge Loss (MIHLSVM)**
TB1o	15	5	**86.67%**	**86.67%**	13.33%
TB1o	15	5	**80.00%**	**80.00%**	60.00%
TB1o	15	5	**93.33%**	**93.33%**	33.33%
TB2o	15	5	53.33%	46.67%	**66.67%**
TB2o	15	5	**53.33%**	**53.33%**	40.00%
TB2o	15	5	**86.67%**	**86.67%**	33.33%
TB1o	20	10	65.00%	65.00%	65.00%
TB1o	20	10	45.00%	45.00%	45.00%
TB1o	20	10	30.00%	30.00%	30.00%
TB2o	20	10	40.00%	40.00%	40.00%
TB2o	20	10	40.00%	40.00%	40.00%
TB2o	20	10	35.00%	35.00%	35.00%
TB1o	25	10	**84.00%**	**84.00%**	40.00%
TB1o	25	10	**96.00%**	**96.00%**	36.00%
TB1o	25	10	**64.00%**	**64.00%**	60.00%
TB2o	25	10	**72.00%**	68.00%	56.00%
TB2o	25	10	**76.00%**	**76.00%**	36.00%
TB2o	25	10	76.00%	76.00%	76.00%

Table 2.2 highlights accuracy differences for the three loss functions. Although separating hyperplanes are different, accuracies are the same in cases with 20 bags and 10 features. When the number of bags increase or the number of features decrease, accuracies tend to change, hard margin usually performing the best among the three. This is more apparent for larger and fuzzier data sets that are presented in Section 2.5.3. It should be noted ramp loss formulation takes significantly more time that hinge and hard margin loss in all test cases, thus it is omitted from further

benchmark problems. The complexity of ramp loss SVM for conventional data is an open problem but we conjecture that multiple instance learning with ramp loss is $\mathcal{NP}$-hard.

2.5.2 Heuristic Performance: Optimal Solution and Time

In order to assess the capabilities of different formulations, we employ principal component analysis (PCA) on the MUSK1 data set so variability of data can be controlled by choosing a subset of features. When controlling the size of the problems, features with larger (smaller) weights in the first few principal components can be selected to create data sets with more (less) variability. This is a naive process that sheds a light on the analysis since data with less variability is typically harder to separate with a separating hyperplane. We use IBM ILOG CPLEX Optimization Studio 12.2 [37] for all exact formulations and set the time limit to 30 minutes. As values greater than 1 do not lead to a significant decrease in the number of misclassifications but an artificial increase in the optimality gap for our heuristic, we set $C = 1$ for our experiments in this section as well.

Tables 2.3, 2.4, and 2.5 show that formulations **IP3** and **CP2** perform the best. In fact, **IP3** is superior to other formulations in a majority of test instances but **CP2** is particularly successful when number of features increase, which makes separation relatively easier. Our results show that, although we consider a harder generalization of an $\mathcal{NP}$-hard problem in MIL context, medium sized problems can be solved in reasonable time using effective formulations.

Our heuristic also performs well compared to the optimal solution in terms of objective function value. It can be observed that the largest difference in objective function value between the heuristic and optimal solution in harder data sets is close to 9, when the total number of instances are 320 and the number of features was 10, which is a difficult separation problem. Although the optimality gap seems to

Table 2.3: Computational results for harder data sets (i.e., subset of MUSK1 with less variability).

		CPU Time (sec.)						Objective Value	
# of Inst.	**# of Feat.**	**IP1**	**IP2**	**IP3**	**CP1**	**CP2**	**3-Phase Heuristic**	**3-Phase Heuristic**	**OPT**
40	10	1.08	0.41	**0.28**	2.65	1.48	**0.01**	3.86	3.58
40	20	0.11	0.22	0.11	0.26	**0.10**	**0.01**	2.00	1.41
40	40	0.11	0.17	0.10	0.07	**0.05**	**0.01**	0.26	0.24
40	80	0.21	0.21	**0.19**	0.21	**0.19**	**0.01**	0.13	0.12
80	10	0.09	0.08	**0.06**	0.48	0.11	**0.01**	1.00	1.00
80	20	2.12	1.18	**1.17**	5.01	3.15	**0.02**	3.86	2.12
80	40	8.81	3.54	**3.44**	6.31	5.02	**0.03**	1.97	1.66
80	80	6.12	4.67	20.07	3.35	**3.27**	**0.02**	0.32	0.26
120	10	156.59	**3.17**	3.27	N/A	475.74	**0.03**	7.13	7.10
120	20	3.91	3.27	**2.21**	N/A	16.30	**0.02**	4.68	3.25
120	40	1218.48	30.51	**21.95**	N/A	N/A	**0.07**	6.83	4.72
120	80	4.01	5.71	3.94	8.56	**3.38**	**0.06**	1.37	0.79
160	10	N/A	15.58	**13.10**	N/A	N/A	**0.10**	11.25	9.75
160	20	N/A	444.97	**295.91**	N/A	N/A	**0.05**	14.39	10.58
160	40	N/A	**47.55**	52.09	N/A	N/A	**0.06**	5.04	4.26
160	80	N/A	29.01	**21.06**	72.51	54.76	**0.12**	2.38	1.59
200	10	N/A	47.39	**43.43**	N/A	N/A	**0.08**	12.85	11.75
200	20	N/A	49.63	**38.06**	N/A	N/A	**0.05**	9.21	7.70
200	40	N/A	**123.63**	132.15	N/A	N/A	**0.07**	4.83	3.79
200	80	N/A	**15.83**	17.11	301.97	47.35	**0.15**	1.48	1.26
240	10	142.76	6.12	**4.10**	N/A	N/A	**0.13**	9.16	9.01
240	20	N/A	464.55	**291.64**	N/A	N/A	**0.08**	11.07	10.49
240	40	N/A	**173.80**	205.40	N/A	N/A	**0.14**	6.74	5.25
240	80	N/A	**1768.32**	N/A	N/A	N/A	**0.21**	5.14	3.60
280	10	N/A	20.90	**8.76**	N/A	N/A	**0.13**	11.95	11.00
280	20	N/A	N/A	N/A	N/A	N/A	**0.13**	20.65	N/A
280	40	N/A	N/A	N/A	N/A	N/A	**0.20**	11.92	N/A
280	80	N/A	1510.73	**899.54**	N/A	N/A	**0.41**	5.28	3.49
320	10	N/A	885.57	**559.06**	N/A	N/A	**0.22**	25.57	16.88
320	20	N/A	N/A	N/A	N/A	N/A	**0.22**	46.24	N/A
320	40	N/A	N/A	N/A	N/A	N/A	**0.24**	14.51	N/A
320	80	N/A	**1602.74**	N/A	N/A	N/A	**0.56**	6.62	3.71
360	10	N/A	N/A	N/A	N/A	N/A	**0.33**	32.99	N/A
360	20	N/A	N/A	N/A	N/A	N/A	**0.20**	23.22	N/A
360	40	N/A	N/A	N/A	N/A	N/A	**0.29**	12.77	N/A
360	80	N/A	1529.58	**1116.26**	N/A	N/A	**0.68**	9.23	3.93
400	10	N/A	N/A	N/A	N/A	N/A	**0.37**	25.41	N/A
400	20	N/A	N/A	N/A	N/A	N/A	**0.19**	34.42	N/A
400	40	N/A	N/A	N/A	N/A	N/A	**0.38**	14.98	N/A
400	80	N/A	N/A	N/A	N/A	N/A	**0.39**	6.24	N/A

be large, it should be noted that 8 or less additional bags are misclassified (among more than 60 bags) compared to the optimal solution with significant time savings.

Table 2.4: Computational results for easier data sets (i.e., subset of MUSK1 with more variability).

		CPU Time (sec.)						Objective Value	
# of Inst.	**# of Feat.**	**IP1**	**IP2**	**IP3**	**CP1**	**CP2**	**3-Phase Heuristic**	**3-Phase Heuristic**	**OPT**
40	10	0.33	0.12	**0.07**	0.45	0.40	**0.02**	4.20	4.00
40	20	0.06	**0.05**	**0.05**	0.11	**0.05**	**0.01**	1.00	1.00
40	40	**0.06**	0.08	0.10	0.26	0.30	**0.01**	1.00	1.00
40	80	0.36	0.43	**0.29**	0.55	0.38	**0.02**	0.75	0.49
80	10	0.11	0.16	**0.07**	1.66	4.63	**0.03**	3.24	3.21
80	20	97.21	2.08	**1.07**	78.36	18.54	**0.04**	7.36	5.87
80	40	1.76	1.71	**1.12**	3.91	1.13	**0.03**	2.72	1.94
80	80	**4.44**	6.34	4.86	9.51	6.74	**0.02**	1.31	1.19
120	10	N/A	2.82	**1.57**	139.29	79.87	**0.04**	11.11	9.05
120	20	N/A	7.23	**4.12**	N/A	639.66	**0.04**	11.02	9.13
120	40	N/A	47.67	**31.89**	N/A	N/A	**0.05**	11.90	6.71
120	80	8.11	**3.51**	9.58	6.21	5.75	**0.06**	0.85	0.85
160	10	N/A	2.75	**1.38**	N/A	997.12	**0.09**	11.34	10.38
160	20	N/A	67.07	**35.90**	N/A	N/A	**0.06**	15.94	12.05
160	40	N/A	**90.21**	91.23	N/A	N/A	**0.07**	8.76	6.37
160	80	1666.50	**23.87**	29.74	N/A	N/A	**0.09**	4.29	3.54
200	10	N/A	9.11	**5.59**	N/A	347.75	**0.12**	14.25	14.22
200	20	N/A	19.19	**14.87**	N/A	N/A	**0.06**	10.12	9.73
200	40	N/A	**103.92**	134.32	N/A	N/A	**0.08**	15.16	9.70
200	80	N/A	**185.59**	194.51	N/A	N/A	**0.20**	7.82	3.93
240	10	55.55	2.87	**1.35**	N/A	N/A	**0.13**	8.77	8.77
240	20	N/A	449.23	**413.07**	N/A	N/A	**0.12**	18.67	15.95
240	40	N/A	**787.16**	1034.43	N/A	N/A	**0.09**	15.50	11.75
240	80	464.77	420.30	**203.53**	N/A	N/A	**0.11**	5.33	4.37
280	10	N/A	11.63	**7.09**	N/A	N/A	**0.21**	14.27	14.25
280	20	N/A	**217.74**	218.41	N/A	N/A	**0.13**	16.67	16.19
280	40	N/A	482.76	**397.70**	N/A	N/A	**0.19**	13.51	10.90
280	80	N/A	**249.66**	434.33	N/A	N/A	**0.21**	7.54	4.30
320	10	N/A	1257.40	**790.38**	N/A	N/A	**0.29**	31.59	30.38
320	20	N/A	372.36	**207.43**	N/A	N/A	**0.24**	17.75	17.49
320	40	N/A	N/A	N/A	N/A	N/A	**0.24**	30.91	N/A
320	80	N/A	N/A	N/A	N/A	N/A	**0.39**	11.77	N/A
360	10	N/A	94.62	**68.36**	N/A	N/A	**0.30**	21.43	21.38
360	20	N/A	744.08	**562.85**	N/A	N/A	**0.31**	20.13	18.96
360	40	N/A	N/A	N/A	N/A	N/A	**0.40**	30.69	N/A
360	80	N/A	N/A	N/A	N/A	N/A	**0.31**	8.52	N/A
400	10	N/A	301.65	**205.37**	N/A	N/A	**0.41**	26.29	26.25
400	20	N/A	**949.36**	1155.72	N/A	N/A	**0.20**	25.67	22.00
400	40	N/A	N/A	N/A	N/A	N/A	**0.24**	19.56	N/A
400	80	N/A	N/A	N/A	N/A	N/A	**0.71**	13.79	N/A

Furthermore, we expect proximity of heuristic hyperplane to the optimal hyperplane, thus a subtle difference in cross validation results.

Table 2.5: Computational results for a subset of instances in MUSK1 data set with all features.

		CPU Time (sec.)						Objective Value	
# of Inst.	# of Feat.	IP1	IP2	IP3	CP1	CP2	3-Phase Heuristic	3-Phase Heuristic	OPT
80	166	10.92	9.32	10.28	4.37	**2.37**	**0.03**	0.34	0.30
120	166	222.70	37.73	306.04	67.37	**19.84**	**0.09**	0.34	0.29
160	166	63.78	49.33	173.77	25.59	**17.98**	**0.14**	0.51	0.45
200	166	N/A	138.59	**105.19**	798.40	195.55	**0.31**	1.42	0.99
240	166	N/A	945.99	**464.65**	N/A	838.98	**0.71**	1.45	1.20
280	166	N/A	659.91	373.44	N/A	**353.75**	**0.36**	0.91	0.79
320	166	N/A	655.65	**414.72**	N/A	478.25	**0.61**	1.72	1.04
360	166	N/A	N/A	N/A	N/A	N/A	**1.36**	3.06	N/A
400	166	N/A	N/A	N/A	N/A	N/A	**1.37**	3.53	N/A

2.5.3 Robust Classification Performance for Larger Data Sets: Cross Validation Results

2.5.3.1 Linear Classification

In this section, we present leave one bag out cross validation results for linear classification using the three-phase heuristic. All instances and features of MUSK1 data are used in computing these results. We also use a set of C values to observe the effect on the performance of our algorithm. As Table 2.6 shows, highest cross validation accuracy of 79.35% is achieved for $C = 1$.

Table 2.6: Leave-one-bag-out cross validation results for MUSK1 data with 476 instances in 92 bags and 166 features.

C	Hard Margin Loss (Heuristic)		Hinge Loss (CPLEX)	
	LOBOCV	CPU Time (sec.)	LOBOCV	CPU Time (sec.)
0.1	**75.00%**	147.30	51.09%	**16.34**
1	**79.35%**	**217.43**	76.09%	1,818,460.63
10	**73.91%**	**321.21**	63.04%	1,816,458.85
100	**77.17%**	**312.66**	70.65%	1,819,085.86

Table 2.6 also shows the performance of our algorithm against hinge loss formulation (i.e., MIHLSVM) that is solved using CPLEX. Accuracy of our heuristic

algorithm for MIHMSVM is consistently higher than MIHLSVM. It should be noted that the time reported in the table is for validation of 92 bags. For a given C value, it usually takes more than 20 days to perform cross validation using hinge loss formulation on CPLEX, whereas our heuristic takes less than 6 minutes.

2.5.3.2 Nonlinear Classification

In order to assess the performance of our heuristic for nonlinear classification, MUSK2 data is considered with a Gaussian radial basis function. Formally, the Gaussian kernel is represented as

$$\mathbf{K}(\mathbf{x}_j, \mathbf{x}_i) = e^{-\frac{\|\mathbf{x}_j - \mathbf{x}_i\|^2}{2\sigma^2}}. \tag{2.20}$$

Table 2.7: Leave-one-bag-out cross validation and CPU time (in seconds) results for MUSK2 data with 6,598 instances in 102 bags and 166 features.

$2\sigma^2$	$C = 0.5$		$C = 1$		$C = 10$		$C = 100$	
	LOBOCV	**CPU Time**	**LOBOCV**	**CPU Time**	**LOBOCV**	**CPU Time**	**LOBOCV**	**CPU Time**
10	60.78%	22,639.80	63.73%	25,304.07	63.73%	23,736.96	63.73%	23,696.82
25	72.55%	25,804.41	79.41%	12,254.31	81.37%	11,228.20	81.37%	11,834.13
50	57.84%	18,956.22	**84.31%**	3,913.35	80.39%	3,461.50	81.37%	3,397.65
100	56.86%	13,245.11	79.41%	2,180.79	82.35%	1,926.41	81.37%	1,956.40
166	52.94%	13,083.71	76.47%	1,899.21	80.39%	1,559.36	79.41%	1,540.54
200	51.96%	12,998.93	79.41%	1,924.97	78.43%	1,507.11	79.41%	1,409.86
500	49.02%	12,837.94	75.49%	2,138.56	77.45%	1,416.91	79.41%	**1,199.26**
1000	49.02%	12,831.96	44.12%	9,764.32	47.06%	9,287.45	47.06%	9,221.11

Different C and σ values are compared and the results are presented in Table 2.7. The default selection in [34] is also considered that sets $2\sigma^2$ equal to the number of features. The best accuracy achieved is 84.31% for $C = 1$ and $\sigma = 5$. It should be noted that $C = 0.1$ is not presented in Table 2.7 because the regularization term outweighs the misclassification term in the objective function and the same cross validation accuracy of 38.24% is obtained for all values of σ. Our results show that the accuracy tends to decrease when σ increases as this converges to a linear separation. The total time spent for cross validation of 102 bags for our heuristic rarely exceeds

an hour for nonextreme values of parameters. It is also noteworthy to mention that the time spent usually reduces with increased C since the misclassification penalty outweighs the quadratic regularization term in the objective function, providing a relatively more tractable problem.

2.5.4 Wind Farm Site Locating

Wind farm site locating is the first phase in the process of building a wind farm. In order to construct a wind farm, specific conditions need to be satisfied. Therefore, a decision maker should choose a site for a wind farm based on a number of factors such as wind speed, wind availability, water temperature, depth of water, pressure, precipitation, wave speed, wave height, and distance to the shore. Each of these factors can be measured in different potential locations for a site. After that, an expert is to decide whether a location is a good candidate for a wind farm or not.

To implement our method, we use a data set from Irish sea which has been provided by 4C Offshore Co. [1]. The data set consists of 74 sensors (instances) spread into 10 site locations. The features we utilize are the location (i.e., latitude and longitude) of the sensor, wind speed, depth of water, and distance to shore. Each bag consists of a set of instances that are in the same neighborhood. For each bag, we know whether it is an ideal location or not through the current status of a future wind farm. We use LOBOCV technique to check the performance of our method. The LOBOCV provides 80% accuracy, which is reasonable considering the limited amount of data we had to train our classifier.

The benefits of this approach is two-fold: (i) it provides a set of rules to determine if a specific location is suitable for a wind farm or not, and (ii) less data is to be collected for future decisions. First benefit helps with a qualitative analysis shedding a light on which of the aforementioned factors are more critical through an analysis of weights for each feature. Second benefit helps in significantly reducing the costs and

time it takes for data collection before a decision is to be made regarding a potential location.

2.6 Summary

In this chapter, we propose a robust support vector machine classifier for multiple instance learning. We show that hard margin loss classifiers provide remarkably better generalization performance for multiple instance data in practice, which is in line with theory. We develop three integer programs and two constraint programs and compare their time performance in achieving optimal solutions. Furthermore, we develop a heuristic that can handle even large problem instances within seconds. Our heuristic provides higher cross validation accuracy for MIL data compared to conventional hinge loss based SVMs in significantly less time. We use wind farm site location data to show the implementation of our approach.

Chapter 3 Offshore Wind Farm Layout Optimization

3.1 Introduction

Wind energy is becoming quite important in many different venues around the world. Its use as an alternative source of clean, reliable, and sustainable energy is making it a seriously considered natural source for production of electricity by many countries. No other alternative energy source has been more successfully implemented than wind energy. There are a number of reasons for those success stories. Wind is abundantly available and never depleted (e.g., day and night). It is clean and the harnessing of wind has only minor side effects on the environment. Those side effects, such as noise, disturbance of the natural view, and the so-called stroboscope effect, are even less considerable in the case of offshore wind farms. Offshore wind turbines are a commonly used power source in European countries that are more densely settled. European Union has recognized the importance of the commitment to renewable energy and subsidizes wind farm companies through the Renewable Energy Law created in 2000.

Even in the United States, where mainly energy comes from fossil fuels and nuclear power, the advantages of wind generated energy have been thoroughly investigated. The goal of producing 20% of the nation's energy demand from wind energy by 2030 is technically feasible, not cost-prohibitive, and provides numerous benefits. Some of these benefits are carbon emission reductions, natural gas fuel savings, and water savings according to the National Renewable Energy Laboratory (NREL).

The increase in global energy demand, especially considering the rapid industrial development of the so-called third world countries, calls for alternative sources of

energy, other than fossil fuels. According to the U.S. Energy Information Administration, the prospective growth of global energy consumption will be 49% from 2007 to 2035, which is equivalent to an increase in energy use from 495 quadrillion British thermal units (BTU) to 590 quadrillion BTU in 2020 and 739 BTU in 2035. To obtain this goal, the cost of wind energy should be reduced to become economically interesting for the use of customers. The optimization framework presented here aims to reduce the cost of energy by optimizing the location of the wind turbines in the offshore wind farm.

Next, in Section 3.2, we briefly review the previous work on wind farm optimization. Section 3.3 defines the problem and describes important elements of our problem such as power curve, wake model, and wind speed model used. In Section 3.4, we present the mathematical formulation of our problem. Section 3.5 presents the computational results. We provide a brief summary in Section 3.6.

3.2 Background

This section explores the previous researches conducted in this line of study. There are many different strategies that have been established in the optimization of onshore wind farms. However, the optimization of offshore wind farms is a developing study, which will produce innovative and improved methods.

Some studies focus on the minimization of wake losses, under the assumption that this will produce the optimal profits for the wind farm through maximum performance. Samorani [75] identifies the wind farm layout optimization problem (WFLOP) as an important aspect in the design of a wind farm. Their stance is congruent with the idea mentioned above. Szafron [83] considers the distribution in a wind farm using only turbine spacing to seek similar results by minimizing the wake effect. Rašuo and Bengin [67] introduce a model that focuses on the same idea, but uses a modified

version of the genetic algorithm to overcome the limitation of binary results, which normally accompany the genetic algorithm. This should enable a team to adjust the turbine positions freely and minimize the wake effect even further. The wake effect must be taken into consideration in any model that attempts to optimize the layout of an offshore wind farm, though this should not stand alone without other considerations.

The cost of energy (COE) is addressed in other studies utilizing some of the conditions that are not addressed in the previously mentioned research. Nandigam and Dhali [62] go into more detail with the factors that impact the optimization of an offshore wind farm layout; electrical system type, farm-topology, transmission voltage, wind turbine type, rated power, wind speed, and transmission length. However, the topography of the land does not influence offshore wind farms as much as onshore. Elkinton et al. [24] include factors such as wind and wave climates, soil conditions, and water depths in their study of the COE relating to offshore wind farms. As the number of factors increases, separating the model's components becomes plausible. Their model is divided into major costs, energy production, and energy losses before implementing heuristic optimization algorithms.

Levelized cost of energy (LCOE) minimization is another strategy used in offshore wind farm optimization. Lackner and Elkinton [46] develop a LCOE function that allows the annual energy production to be modeled as a function of turbine position only. As with most optimization models, the wind speed probability distribution function is approximated by a Weibull distribution. Kusiak and Song [45] take a similar stance by utilizing the LCOE with a slight adjustment. Their objective function includes an added parameter, the levelized replacement cost. Mahat et al. [51] consider the minimization of the real power loss as their objective function. This methodology enables them to relate the transmission loss to the reactive power, which is consumed by the turbine. The group combines this minimization equation with the

hereford ranch algorithm to obtain the optimal distribution generation. Minimizing this function yields the amount of real power that the wind turbine has to produce at various locations minimizing the real loss. There are many external conditions that affect the optimal layout for an offshore wind farm; primarily, floating ice, wind, and waves are primary external conditions [53]. These external conditions, among others such as water depths and soil conditions [46], can be countered through design or data estimation to minimize their affect on the performance and cost of an offshore wind farm [53]. The relative cost of production moving from an offshore wind farm to an onshore wind farm is estimated to increase 30-60% [62]. Fuglsang and Thomsen [28] show a 28% increase in the annual production of energy when comparing an offshore wind farm to an onshore wind farm, which is caused by the increase in wind climate from onshore to offshore. This illuminates the importance of exploiting the higher wind speeds available for offshore wind farms.

Optimization in wind farm planning is a balance between the maximum performance and minimum cost in a wind farm layout. The wake effect plays an important role in wind farm planning, as was previously discussed, which is represented in [75] and [83] among others. Samorani [75] also accounts for construction and logistics factors in the solution to the WFLOP. Nandigam and Dhali [62] apply geometric programming as a tool to configure optimal layouts based on cost, loss, and reliability models. Optimization methods have been widely used to find the optimal layout solution for a wind farm. Especially for onshore farms, there are different software packages available to specify the most profitable layout. The most commonly used algorithms are the genetic algorithm and the greedy heuristic algorithm. Serrano Gonzalez et al. [77] define a mixed integer problem to optimize the profits of an offshore wind farm. The problem is developed with net cash flow and initial capital investment, and an evolutionary algorithm is used to solve the mixed integer problem. Other researchers expanded the genetic algorithm by adding mobility (add, remove,

move) options or performing other meta-heuristic approaches, like the simulated annealing procedure to overcome some of the limitations of the search algorithm[70].

Through all of these studies, there are not many constraints that have been considered for offshore wind farm optimization models. Minimum distance constraints forbid turbines to be installed too close to each other. Constraints on the number of turbines and boundaries for power generated have also been considered. It appears that a total comprehensive model, which includes every important constraint, has not yet been established. There are some studies on onshore wind farm layout optimization (see e.g., [75, 24, 46, 67, 51, 45, 70]), but it should be noted that there are major differences in objectives and limitations of offshore and onshore wind farms. Samorani [75] presents a complete review based on wind farm layout optimization both onshore and offshore. Another study on offshore wind farm layout optimization is [70] but, it doesn't define the complete mathematical model for the layout optimization problem. It should also be noted that these studies consider discrete space for wind turbine locations that may lead to suboptimal solutions although, it can be argued that the optimality gap would not be significant. The literature review shows there is potential for improvement on offshore wind farm layout optimization based upon mathematical formulation of the problem and the solution methods. Next, we define problem and explain the important elements of it.

3.3 Problem Description

There are two main questions with respect to wind farm turbine positioning problem. The first one is how many turbines should be placed on the farm and the second one is where these turbines should be located. A natural objective may be to maximize the profit for the farm, considering the cost of construction and the revenue stream from future energy production. We consider the cost function defined by

Mosetti et al. [59] for installation, which is proportional to the number of turbines installed. We therefore, consider the problem of placing a fixed number of turbines and maximizing the power production. Next, this problem is solved by changing the number of turbines to find a suitable number of them to place within the site.

There is a wide range of additional considerations that may have to be taken into account at a given site for a wind farm. These considerations may include design of the electrical connection system, impact on wildlife and other environmental effects. We do not consider such site-specific considerations in our model; however, we show how a set of possible turbine positions can be excluded if such considerations render them infeasible.

Next, we review the operating characteristics of a wind turbine and the relationship between power output and wind speed. We then discuss how wind turbines reduce the power output of turbines placed downwind (i.e., *wake effect*).

3.3.1 Power and Thrust Curve

The power generated by a wind turbine is directly related to the wind speed and its direction distributions at hub height. Since the wind does not blow uniformly and from one direction, different wind directions will have different wake effects. Those wind turbines that are placed upwind result in lower wind speeds at the downwind turbines. Furthermore, wind speed varies with distance to shore, so each turbine in a farm typically experiences a different wind speed.

A wind turbine needs a minimum wind speed to start operating (*cut-in*). The power output then increases with the wind speed until the nominal or rated power of the turbine is reached. The blades on the turbine are then regulated (*pitch control*), such that the power output remains the same with increasing wind speed until a maximum wind speed is reached. When this wind speed is reached, the turbines are stopped (*cut-out*) to prevent damage to the blades and the support structure. The

power curve is provided as a function of speed, either continuously or for a set of speed values, in the manufacturers' technical sheets. In the latter case, it is necessary to use an interpolation technique so that the power curve is defined for every speed. Total power generation for all the turbines in the wind farm can be obtained as the summation of the power generation of each turbine.

Another important characteristic of a wind turbine generator is the thrust coefficient curve, which depicts the relationship between the power produced and the value of the thrust coefficient at any wind speed between the cut-in and cut-out. The thrust coefficient measures the proportion of energy captured when the wind passes through the blades of the turbine [2]. It reaches its maximum when the turbine first reaches its nominal power and then decreases with increased wind speed when the blades are pitched.

3.3.2 Wake Effect Model

Turbine wakes describe the area behind the rotating wind turbine blades, where the wind speed is influenced by the motion of the rotor blades. Usually, the wind speed within a wake is decreased, which effects the energy production of the affected turbine considerably. Also, the wind in the wake is more turbulent and can therefore, speed up the process of material fatigue or increase the stress on the material. The issue of turbine wake needs to be considered in a large wind farm, since more turbines generate more power; but when they are placed too closely to each other, the wake effect can outweigh the higher energy generation. Multiple models have been created to deal with the wake effect for calculating an appropriate penalty for placing turbines inside a wake. Usually, some assumptions are being made, such as uniform incoming wind speed and a linear expanding cone behind the turbine. Then, the commonly used equation computes the velocity deficit created by the rotor blades. However, this only addresses the wind speed and so far, the wind turbulences have not been

considered in any model. One of the most commonly used wake model in literature is the Jenson model [39]. Although this model does not perfectly model the wake behind a turbine and the interaction of multiple wakes, recent studies have shown that the model gives reasonable approximations of wind speed reductions in small and medium wind farms for distances more than three turbine diameters downwind [6]. More accurate models based on computational fluid dynamic (CFD) codes also exist but, the computational effort required makes them ill-suited for optimizing the layout of wind farms. As a result, all previous works on the optimization of wind farm layouts have used the models described in [39, 41]. Figure 3.1 shows the basic concept of wake behind a wind turbine.

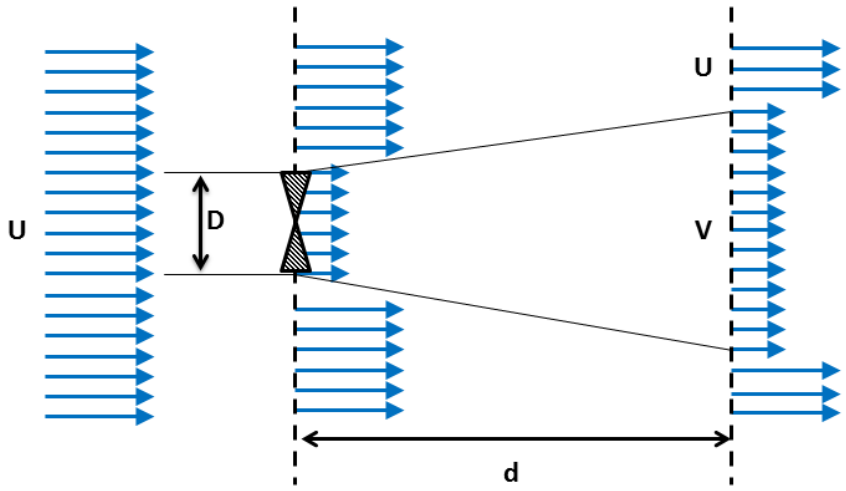

Figure 3.1: Jensen's wake effect model.

It is important to keep in mind that here, the aim is not to accurately predicting the power output of a wind farm, but rather finding suitable layouts. Thus, as long as the models used provide appropriate penalties for placing turbines inside a wake, they preserve ranking of the solutions and can, therefore, be used to optimize the layouts. To get more accurate predictions of the power output from our layouts, we evaluate the solutions using a more comprehensive and computationally demanding

model for the wakes. For a detailed review of different types of wake models, we refer to [18, 92].

The equation for the velocity deficit of this model is as follows:

$$def_{ij} = \frac{1 - \sqrt{1 - C_t}}{(1 + \frac{2\kappa d_{ij}}{D})^2} = 1 - \frac{V}{U}, \tag{3.1}$$

where V is wind speed in the wake of the turbine, U is free stream wind speed, C_t is thrust coefficient of the turbine and depends on incoming wind speed, κ is wake spreading constant, d_{ij} is the distance between turbine j and i projected on the wind direction θ, and D is turbine rotor diameter.

It may be the case that only a part of a turbine is affected by a wake from another turbine while the rest of the turbine sits in the free stream wind speed. We make the simplifying assumption that if the turbine is at least partially inside a wake, the power production of the turbine is equivalent to the power produced at the wind speed in the wake (i.e., the wake covers the whole turbine).

Usually, when multiple turbines create a wake, the root-sum-square of each of the individual velocity deficits is the total velocity deficit on the one turbine affected. Therefore, the velocity deficit resulted from each upstream turbine can be obtained separately via (3.1), then (3.2) will be utilized to compute the total velocity deficit seen by the downstream turbine i [59],

$$Total\ Velocity\ Deficit_i = \sqrt{\sum_{j=1, j\neq i}^{T} def_{ij}^2}. \tag{3.2}$$

It should be mentioned that not all the turbines generate wake effect at turbine i for a given direction θ. Figure 3.2 shows a wind farm consisting of 4 turbines. Given a wind direction, turbine B is not affected by wake effect of turbine A, but turbine C is affected by wake effects of both turbine A and B. Turbine D is affected only by wake effect of turbine A. Kusiak and Song [45] suggest a method to find wake effects of the

turbines affecting turbine i. In this method, total velocity deficit will be calculated from the following equation:

$$Total\ Velocity\ Deficit_i = \sqrt{\sum_{j=1, j\neq i, \beta_{ij}<\alpha}^{T} def_{ij}^2} \tag{3.3}$$

where α is calculated as $\arctan(\kappa)$ and β_{ij}, which is the angle between two vortex of turbine i and j, is calculated as follows:

$$\beta_{ij} = \arccos\left[\frac{(x_i - x_j)\cos\theta + (y_i - y_j)\sin\theta + \frac{D}{2\kappa}}{\sqrt{(x_i - x_j + \frac{D}{2\kappa}\cos\theta)^2 + (y_i - y_j + \frac{D}{2\kappa}\sin\theta)^2}}\right] \tag{3.4}$$

Finally, if wind turbine i is inside the wake of turbine j, d_{ij} is calculated as $|(x_i - x_j)\cos\theta + (y_i - y_j)\sin\theta|$.

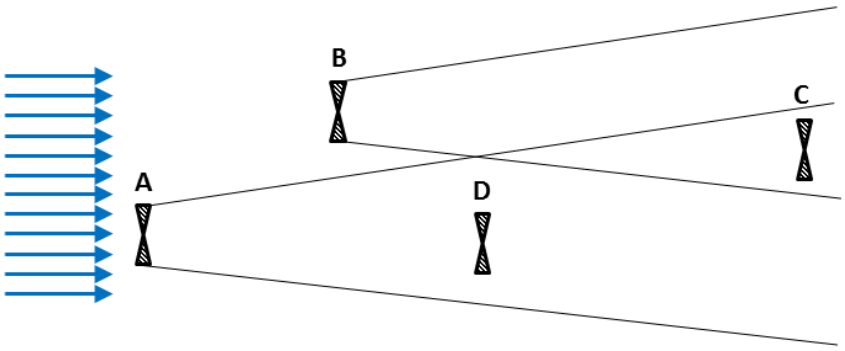

Figure 3.2: Turbine affected by other turbines' wake effect.

3.3.3 Wind Speed and Direction

The wind resource is the single, most important consideration in choosing a site for a wind farm [4]. The prediction of wind resources at a site is often based on a combination of measurements on-site over a time period, such as a year and longer

time series of wind measurements from nearby meteorological stations. The prediction of wind resources, both speed and direction, is a complex task with significant uncertainty. The wind characteristics may change during the 20-year design life of wind farms due to long-term weather changes or changes in the surrounding area. In this work, we assume a representative set of measurements of wind speed and direction is available for the wind farm site we consider.

3.4 Mathematical Model

In this section, we first explain the cost function and continue with formulating the problem with power maximization objective as we discussed in previous sections. Based on [59], the wind farm total turbine cost per year ($cost_{tot}$) function that we use is

$$cost_{tot} = cost_{ty} \times T(\frac{2}{3} + \frac{1}{3}e^{-0.00174T^2}), \tag{3.5}$$

where T represents the total number of turbines placed in the wind farm and $cost_{ty}$ represents the cost of each turbine per year. In this model, the total cost is only dependent on number of turbines installed in the farm and considers some discount when a large number of wind turbines is purchased.

The power generation from wind farm depends on wind distribution and turbine characteristics. The wind distribution is the set of pairs (scenario, probability) that describe the characteristics of the wind in the site under consideration. A scenario s is composed of a direction D_s, i.e., the direction from which the wind blows, and an undisturbed wind speed U_s, i.e., the speed of the wind before reaching the wind farm. Throughout this chapter, the wind speed is always expressed in m/s and the direction as a number in the interval [0, 360), which indicates the angle formed by the direction of the wind and the x-axis (e.g., 0^o is wind from East, 90^o is wind from

North, etc.). The probability r_s associated with a scenario s is the probability of realization of s, where the wind blows from direction D_s at a speed of U_s m/s.

The turbine characteristics include its physical characteristics that are needed for the computation of the wake effect generated by that turbine (hub height, rotor diameter, and the thrust coefficient curve) and its power curve, i.e., the function P_v that computes the power generated given the wind speed at the turbine. Given this information, it is possible to estimate the expected power produced by averaging the power generated under every scenario by weighting each term by the probability of realization of the corresponding scenario:

$$P = \sum_{s \in S} r_s P_s = \sum_{s \in S} r_s \sum_{i \in P} P_v(v_i^s) = \sum_{s \in S} r_s \sum_{i \in P} P_v(U_s \cdot [1 - \sqrt{\sum_{j \in w_i^s} (def_{jis})^2}]). \qquad (3.6)$$

The power P_s generated under scenario s is equal to the sum of the power generated by the individual turbines under scenario s. The power generated by a turbine i under scenario s is the power corresponding to speed v_i^s in the power curve, where v_i^s is the wind speed at the turbine position under scenario s. As seen in the previous section, v_i^s depends on which turbines create a wake that affects i (these turbines form set W_i^s). In our implementation, a turbine j induces a positive velocity deficit on a turbine in position i if the wake created by turbine j intersects the rotor of turbine i. Furthermore, the value of the velocity deficit suffered by turbine i does not depend on the portion of the area swept by its blades that is affected by the wake. In other words, the value of the velocity deficit is the same regardless of whether the rotor of a turbine is fully or partially covered by a wake. We use this conservative wake modeling because previous studies showed that Jensen's model tends to underestimate the velocity deficits within large wind farms. As seen in the previous section, the value of def_{jis} depends also on the wind speed at turbine j, which determines the value of

parameter C_t. From an implementation point of view, it is necessary to compute the wind speeds at turbine positions following the same order in which the wake effects are applied, i.e., first compute the speed at the turbines that are not affected by any wake, then the speed at those turbines affected only by wakes generated by the former set of turbines.

Let us assume that the available area is a rectangle of size $g^1_{max} \times g^2_{max}$. Each turbine is associated with 2 continuous variables (g^1_i and g^2_i) representing its coordinates. A set of binary variables y_{jis} is needed to indicate if turbine j creates a wake effect on turbine i under scenario s. $y_{jis} = 1$ if turbine j, creates a wake effect on turbine i under scenario s, and $y_{jis} = 0$ otherwise. A set of continuous variables def_{jis} are set equal to the velocity deficit induced by turbine j on turbine i under scenario s and under the assumption that the wake generated by j actually affects i. This assumption is implemented by multiplying the terms def^2_{jis} to the binary variables y_{jis} in the objective function. In light of these assumptions, the model that maximizes

the expected power generation is the following:

$$\max \quad \sum_{s\in S} r_s \sum_{i\in P} P_v(U_s \cdot [1 - \sqrt{\sum_{j\in w_i^s} (def_{jis})^2 \cdot y_{jis}}]) \tag{3.7a}$$

$$\text{subject to} \quad (g_i^1 - g_j^1)^2 + (g_i^2 - g_j^2)^2 \geq (3D)^2 \qquad \forall i,j \in P, j \neq i \tag{3.7b}$$

$$y_{jis} = \begin{cases} 1 & \text{if } j \in w_i^s \\ 0 & \text{otherwise} \end{cases} \qquad \forall i,j \in P, j \neq i, s \in S \tag{3.7c}$$

$$def_{jis} = \frac{1 - \sqrt{1 - C_t}}{1 + 2\kappa(\frac{\sqrt{(g_i^1 - g_j^1)^2 + (g_i^2 - g_j^2)^2}}{D})} \qquad \forall i,j \in P, j \neq i, s \in S \tag{3.7d}$$

$$0 \leq g_i^1 \leq g_{max}^1 \qquad \forall i \in P \tag{3.7e}$$

$$0 \leq g_i^2 \leq g_{max}^2 \qquad \forall i \in P \tag{3.7f}$$

$$y_{jis} \in \{0,1\} \qquad \forall i,j \in P, j \neq i, s \in S \tag{3.7g}$$

$$def_{jis} \geq 0 \qquad \forall i,j \in P, j \neq i, s \in S. \tag{3.7h}$$

In this model, constraint (3.7b) enforces the proximity constraint discussed in the previous section. Constraint (3.7c) represents the set of constraints needed to verify if turbine j creates a wake effect on turbine i under scenario s, and therefore, it should express the operations needed to perform this task. This is possible, for example, by (i) applying a roto-translation of the axis so that the wind blows from East and turbine j is in (0,0) and (ii) comparing the position of the rotor of i to the point where the wake generated by j intersects the line $x = g_i^1$. Since this is not the model we use, we prefer keeping a compact view of the constraints represented by (3.7c). Constraint (3.7d) computes the value of def_{jis} according to Jensen's model. Note that the model is impractical because it is highly nonlinear.

Alternatively, we can consider a finite set of candidate positions, each one associated with a binary variable x_i whose value is 1 if a turbine is present in position i,

and 0 otherwise. This model has the advantage of computing the velocity deficit for every pair of turbines and every scenario s during the preprocessing phase. In this case, we introduce continuous variables associated with the wind speed (v_i^s) and the power generated (p_i^s) at each position for each scenario. The model that maximizes the expected power generation for a finite set of candidate positions is as follows:

$$\max \quad \sum_{s \in S} r_s \cdot \sum_{i \in P} P_v(v_i^s) \cdot x_i \tag{3.8a}$$

$$\text{subject to} \quad 1 - x_i \geq x_j \qquad \forall i \in P, j \in N(i), j \neq i \tag{3.8b}$$

$$v_i^s = U_s \cdot [1 - \sqrt{\sum_{j \in w_i^s} (def_{jis})^2 \cdot x_j}] \qquad \forall i, j \in P, j \neq i, s \in S \tag{3.8c}$$

$$\sum_{i \in P} x_i = T \tag{3.8d}$$

$$x_i \in \{0, 1\} \qquad \forall i \in P \tag{3.8e}$$

$$v_i^s \geq 0 \qquad \forall i \in P, s \in S. \tag{3.8f}$$

Constraint (3.8b) forbids to place a turbine in position j if a turbine is present in a neighboring position i. The neighborhood of a position i, which we call $N(i)$, is the set of all positions whose distance to i is less than 3 rotor diameters. Constraint (3.8c) computes the values of the variables v_i^s. Note that the set W_i^s, as well as the values of the velocity deficits, are known a-priori because the number of positions considered is finite. We need to include constraint (3.8d) to install exactly T turbines.

There are two reasons this model cannot be solved very efficiently: (i) Nonlinearity of typical power functions that is to be used in the objective function, and (ii) Nonlinearity of constraint (3.8c). Even though it may be possible to find a suitable P_v, or approximate it with a piecewise linear function, the widely used wake model in the literature that is presented is quadratic. Therefore, we first attempt to linearize constraint (3.8c) using Newton's square root method, which estimates the square root

of a positive number N as

$$\sqrt{N} \approx \frac{1}{2}(\frac{N}{E} + E), \tag{3.9}$$

where E is an educated guess. Since $\sum_{j \in w_i^s} (def_{j,i,s})^2 \cdot x_j$ never exceeds 1, we consider separating $[0, 1]$ into n intervals to make better educated guesses. Furthermore, we observe that guesses become even more important for smaller values of N. Therefore, we define intervals in an exponentially increasing fashion. Assuming $b_0, \ldots, b_n$ with $b_0 = 0$ and $b_n = 1$ are the break points in $[0, 1]$, we ensure $b_{k+1} - b_k = 2(b_k - b_{k-1})$ and educated guess $E_k = \sqrt{(b_k + b_{k-1})/2}$.

To use Newton's square root approximation, a new set of binary variables (z_{ik}^s) has been introduced to find the appropriate educated guess that should be used to

approximate $\sum_{j \in w_i^s} (def_{jis})^2 \cdot x_j$. The new formulation is as follows:

$$\max \quad \sum_{s \in S} r_s \cdot \sum_{i \in P} p_v(v_i^s) \cdot x_i \tag{3.10a}$$

$$\text{subject to} \quad 1 - x_i \geq x_j \qquad \forall i \in P, j \in N(i), j \neq i \tag{3.10b}$$

$$b_{k-1} - (1 - z_{ik}^s) \leq \sum_{j \in w_i^s} (def_{jis})^2 \cdot x_j \qquad \forall i \in P, s \in S, k \in K \tag{3.10c}$$

$$\sum_{j \in w_i^s} (def_{jis})^2 \cdot x_j \leq b_k + (1 - z_{ik}^s) \qquad \forall i \in P, s \in S, k \in K \tag{3.10d}$$

$$\sum_{k \in K} z_{ik}^s = 1 \qquad \forall i \in P, s \in S \tag{3.10e}$$

$$\sum_{i \in P} x_i = T \tag{3.10f}$$

$$v_i^s \leq U_s \cdot [1 - \frac{1}{2}(\frac{\sum_{j \in w_i^s} (def_{jis})^2 \cdot x_j}{\sum_{k \in K} z_{ik}^s \cdot E_k} + \sum_{k \in K} z_{ik}^s \cdot E_k)] \qquad \forall i \in P, s \in S \tag{3.10g}$$

$$x_i \in \{0, 1\} \qquad \forall i \in P \tag{3.10h}$$

$$z_{ik}^s \in \{0, 1\} \qquad \forall i \in P, s \in S, k \in K \tag{3.10i}$$

$$v_i^s \geq 0 \qquad \forall i \in P, s \in S. \tag{3.10j}$$

The non-linearity of constraint (3.10g) is eliminated by multiplication of $\sum_{k \in K} z_{ik}^s \cdot E_k$ to both sides of the constraint, introducing a new set of continues variables (vz_{ik}^s) instead of ($v_i^s \cdot z_{ik}^s$) and constraints (3.11h), (3.11i), and (3.11j). It should be noted that $[(\sum_{k \in K} z_{ik}^s \cdot E_k) \cdot (\sum_{k \in K} z_{ik}^s \cdot E_k)]$ is equal to $(\sum_{k \in K} z_{ik}^s \cdot E_k^2)$ since (i) for each i and s only one of the z_{ik}^ss are equal to one ($z_{ik1}^s \cdot z_{ik2}^s = 0, k1 \neq k2, \forall i, s$), and (ii) the square of one or zero is equal to one or zero respectively ($z_{ik}^s \cdot z_{ik}^s = z_{ik}^s, \forall i, k, s$). The

formulation with linearized constraints is as follows:

$$\max \quad \sum_{s\in S} r_s \cdot \sum_{i\in P} p_v(v_i^s)\cdot x_i \tag{3.11a}$$

$$\text{subject to} \quad 1 - x_i \geq x_j \qquad \forall i \in P, j \in N(i), j \neq i \tag{3.11b}$$

$$b_{k-1} - (1 - z_{ik}^s) \leq \sum_{j\in w_i^s} (def_{jis})^2 \cdot x_j \qquad \forall i \in P, s \in S, k \in K \tag{3.11c}$$

$$\sum_{j\in w_i^s} (def_{jis})^2 \cdot x_j \leq b_k + (1 - z_{ik}^s) \qquad \forall i \in P, s \in S, k \in K \tag{3.11d}$$

$$\sum_{k\in K} z_{ik}^s = 1 \qquad \forall i \in P, s \in S \tag{3.11e}$$

$$\sum_{i\in P} x_i = T \tag{3.11f}$$

$$\sum_{k\in K} vz_{ik}^s \cdot E_k \leq U_s \cdot [\sum_{k\in K} z_{ik}^s \cdot E_k - \frac{1}{2}(\sum_{j\in w_i^s} (def_{jis})^2 \cdot x_j + \sum_{k\in K} z_{ik}^s \cdot E_k^2)] \qquad \forall i \in P, s \in S \tag{3.11g}$$

$$vz_{ik}^s \geq v_i^s - U_s(1 - z_{ik}^s) \qquad \forall i \in P, s \in S, k \in K \tag{3.11h}$$

$$vz_{ik}^s \leq v_i^s \qquad \forall i \in P, s \in S, k \in K \tag{3.11i}$$

$$vz_{ik}^s \leq U_s \cdot z_{ik}^s \qquad \forall i \in P, s \in S, k \in K \tag{3.11j}$$

$$x_i \in \{0, 1\} \qquad \forall i \in P \tag{3.11k}$$

$$z_{ik}^s \in \{0, 1\} \qquad \forall i \in P, s \in S, k \in K \tag{3.11l}$$

$$v_i^s \geq 0 \qquad \forall i \in P, s \in S \tag{3.11m}$$

$$vz_{ik}^s \geq 0 \qquad \forall i \in P, s \in S, k \in K. \tag{3.11n}$$

This formulation has linear constraints and nonlinear objective function that can be solved using commercially available solvers. A very important feature of this formulation is that any suitable power curve function can be used. Next, we use available datasets that are used in the literature to show the validity of our model

and derive practical conclusions.

3.5 Computational Experiments

Introduced by Mosetti et al. [59], the data we use feature one type of turbine having hub height $h = 60m$, diameter $D = 40m$, and constant thrust coefficient $C_t = 0.88$. The power curve is expressed as follows:

$$P(U) = \begin{cases} 0, & U < 2 \\ 0.3U^3, & 2 \leq U < 12.8 \\ 629.1, & 12.8 \leq U < 18 \\ 0, & 18 \leq U \end{cases} \tag{3.12}$$

where the wind speed U is expressed in m/s and the power in KW. We linearized this power curve as a piecewise linear function using additional binary variables. Results are obtained using IBM ILOG CPLEX Optimization Studio 12.4 [37].

We use two problem instances from Mosetti et al. [59] for this study, both of which consist of finding the optimal layout in a 10×10 square-grid. Each cell is 5D wide and may or may not have a turbine installed at its center. The instances differ in their wind characteristics. In instance 'A', the wind constantly blows from North at 12 m/s, whereas in 'B', the wind speed is 12 m/s, but the direction is uniformly distributed across 36 directions of 10^o each. The wake spreading constant κ is site dependent, and we use a value of 0.0944.

For each problem instance, we run our formulation repetitively increasing the number of turbines (T) and find the cost and maximum power generated. The cost associated with T turbines is calculated using equation 3.5. Next, we find cost per power generated and illustrate the layout for the best ratio. It should be noted that for smaller T values, there might be alternative optimal solutions but we use the one

that is provided by [37]. Next, we find cost per power generated ratio and illustrate the layout for the best ratio. As discussed in Section 3.4, this will result in the best wind farm layout under aforementioned assumptions, which would be very useful for further sensitivity analysis shedding a light on practical questions.

Table 3.1: Result for problem instance A.

T	**Power (KW)**	**Cost**	**Cost per power** ($\times\$10^{-3}/KW$)
21	12042.6	1.724972426	1.43239203
22	12598.1	1.782571292	1.414952487
23	13153.6	1.838724086	1.397886576
24	13709.1	1.893645051	1.381305156
25	14264.6	1.947548405	1.3653018
26	14820.1	2.000644777	1.349953629
27	15375.6	2.053137992	1.335322194
28	15931	2.105222267	1.321462725
29	16486.5	2.157079836	1.308391615
30	17042	2.20887903	1.296138381
31	17562.9	2.260772803	1.287243452
32	18083.8	2.312897712	1.27898877
33	18604.7	2.365373319	1.271384822
34	19125.6	2.418301986	1.264431958
35	19646.5	2.471769033	1.25812182
36	20167.4	2.525843219	1.252438697
37	20688.3	2.580577481	1.247360818
38	21209.2	2.636009914	1.242861548
39	21730.1	2.692164917	1.238910505
40	22250.9	2.749054465	1.235480122
41	22723.1	2.80667948	**1.235165747**
42	23195.4	2.865031223	1.235172156
43	23654.5	2.924092716	1.236167628
44	24139.1	2.983840111	1.236102469
45	24598.9	3.044244027	1.23755291
46	25084.2	3.105270798	1.237938941
47	25556.4	3.166883623	1.239174384
48	26015.5	3.229043622	1.241199909
49	26500.8	3.291710765	1.242117508
50	26973	3.354844688	1.243778848

Table 3.1 shows the result for problem instance 'A'. As we increased the number of turbines, the best ratio comes up at 41 turbines for the wind farm. Table 3.2 shows

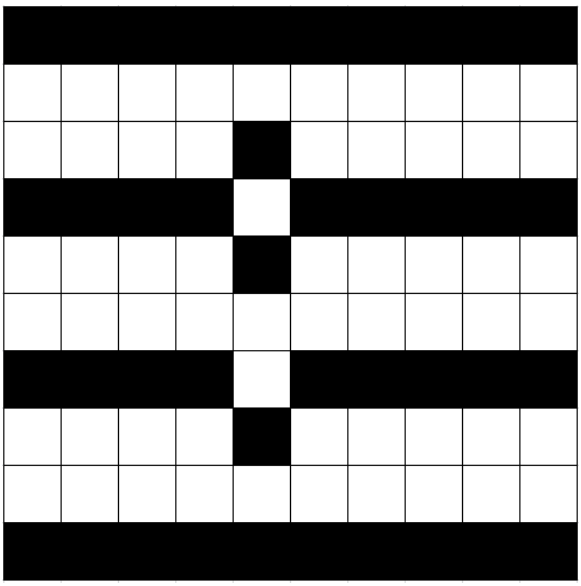

Figure 3.3: Position of turbines for problem instance A.

the result for problem instance 'B'. The best ratio is for 44 turbines in this case. The layouts associated with each problem are illustrated in Figure 3.3 and 3.4. In each layout the turbines are located at the middle of black cells.

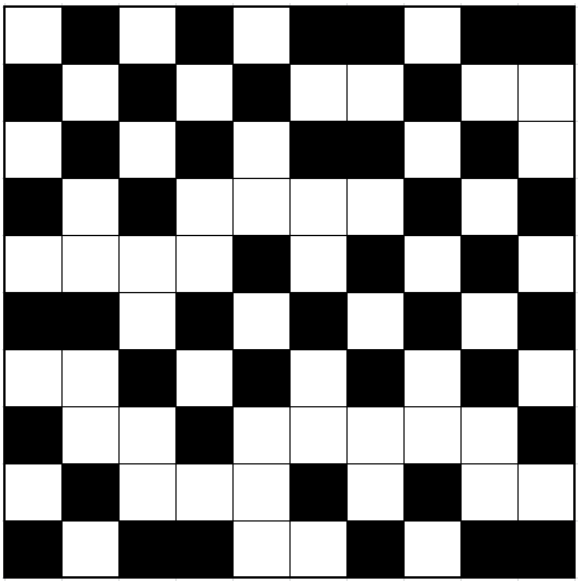

Figure 3.4: Position of turbines for problem instance B.

3.6 Summary

This chapter presents a mathematical model that would efficiently find the optimal layout of turbines in an offshore wind farm. We consider maximizing energy production for formulation purposes but eventually we minimize the cost of energy. We utilize linearization tools that are not only useful under the selected set of assumptions, but for a wide-variety of wind scenarios discussed in the literature. We examine the trade-off between advantages of packing the turbines close together and the loss generated by wake effects. We validate our formulation using Mosetti's problem instances and find the cost per generated power with increasing the number of turbines located.

Table 3.2: Result for problem instance B.

T	Power (KW)	Cost	Cost per power ($\times\$10^{-3}/KW$)
21	11793	1.724972426	1.462708747
22	12280.1	1.782571292	1.451593466
23	12936	1.838724086	1.421400809
24	13316.6	1.893645051	1.422018421
25	13957.8	1.947548405	1.395311873
26	14341.4	2.000644777	1.395013581
27	14919.6	2.053137992	1.376134744
28	15553	2.105222267	1.353579545
29	16052.7	2.157079836	1.343748925
30	16678.5	2.20887903	1.324387103
31	17166.9	2.260772803	1.316937131
32	17685.8	2.312897712	1.307771044
33	18049.8	2.365373319	1.310470653
34	18618.1	2.418301986	1.298898376
35	19136.4	2.471769033	1.291658323
36	19649.5	2.525843219	1.285449105
37	20405.1	2.580577481	1.264672793
38	20852.6	2.636009914	1.264115705
39	21312.3	2.692164917	1.263197739
40	21846.1	2.749054465	1.258373103
41	22256.9	2.80667948	1.261037916
42	22713.7	2.865031223	1.261367027
43	23296.5	2.924092716	1.255163958
44	23965.3	2.983840111	**1.245066872**
45	24369.9	3.044244027	1.249181994
46	24884.5	3.105270798	1.247873495
47	25139.9	3.166883623	1.259704145
48	25671.3	3.229043622	1.257841879
49	26371.5	3.291710765	1.248207635
50	26750.1	3.354844688	1.254142858
51	27406.9	3.41840539	1.247279112
52	27848.7	3.482353812	1.250454711
53	28428.2	3.54665231	1.247582439
54	28687.8	3.611265015	1.258815599
55	29287	3.676158114	1.255218395
56	29593.4	3.741300027	1.264234602
57	30206.3	3.80666153	1.260221056
58	30555.9	3.872215795	1.267256338
59	31164.7	3.937938392	1.263589379
60	31467	4.003807232	1.272382888

Chapter 4 Causal Inference for Time-series Analysis: Simultaneous Denoising and Feature Selection

4.1 Introduction

In this chapter we present the new algorithm that is capable of simultaneously removing the noises while doing the regression. We use this algorithm to find the features contributing most on a response variable. This chapter is organized as follows: In Section 4.2, we investigate the previous works have been done in denoising, feature selection, and regression. In Section 4.3 we first provide the basics of traditional ε-insensitive regression problems and then present our linear and nonlinear formulation for the problem. Next, in Section 4.4 we present our solution algorithm for feature selection. A two-phase heuristic algorithm has been developed for large-scale problems in this section. Section 4.5 demonstrates the computational results for our algorithm performance on natural gas pricing data. We provide a brief summary in Section 4.6.

4.2 Literature Review

This section explores the previous research conducted in this line of study. There are many regression techniques that have been established for the filtering of noise in data and feature selection. Regression is a statistical learning technique which develops a mathematical formula that fits the data. Regression can be used for hypothesis testing, forecasting, inference, and modeling of relationships.

ε-insensitive regression is an optimization based framework for solving regression

problems. It utilizes statistical learning theory and obtains a good generalization from limited size data sets (see [91]). The objective is to optimize a certain boundary to the optimal regression line, therefore, errors within a certain distance (ε) of predicted value are disregarded.

This form of regression is called ε-insensitive because any point in the ε of the anticipated regression function does not contribute to error. An important advantage for considering the ε-insensitive loss function is the sparseness of the dual variables. Representing the solution by a small subset of training points exhibits computational advantages. Furthermore, ε-insensitive regression ensures the existence of a global minimum and minimization of a reliable generalization error bound (see [19]).

ε-insensitive regression has various applications in numerous technology (see e.g., [74], [7]), analytical (see e.g., [49], [36]), and scientific fields (see e.g., [81], [102]). Wu et al. [99] perform location estimation using the Global System for Mobile communication (GSM) based on an ε-insensitive approach which demonstrates promising performances, especially in terrains with local variations in environmental factors. ε-insensitive regression method is also used in agricultural schemes in order to enhance output production and reduce losses (see e.g., [100], [50], [65], [16]). Based on statistical learning theory, ε-insensitive regression has been used to deal with forecasting problems. Performing structural risk minimization rather than minimizing the training errors, the algorithm has better generalization ability than the conventional artificial neural networks (see [33]).

Feature Selection (FS) techniques separate the relevant and non-relevant features in a given model. With the inherent dimensional growth of problems today due to the rapid development of research and technology, the feature selection methods currently developed are proving extremely useful [26]. There are many different techniques in use, but the most commonly implemented techniques are the filter (univariate and multivariate), wrapper (deterministic and randomized), and embedded technique [72].

Grandvalet and Canu [30] introduce an adaptive scaling technique of FS used in face pattern recognition. FS is especially useful in models with datasets containing large numbers of variables [31, 43]. Yang and Pedersen [103] and Ferri et al. [25] evaluate different FS methods. Saeys et al. [73] provide evidence that ensemble feature selection techniques, unification of multiple FS techniques together, generates more robust outcomes than employing only one such technique. Fodor [26] explores linear and non-linear methods to reducing the dimensions in a dataset through feature selection.

Denoising helps to reduce the amount of outliers in optimization as shown in [80]. There are many applicable methods to denoising that are currently in use. Kohler and Lorenz [42] test such methods as the moving average, exponential smoothing, linear fourier smoothing, nonlinear wavelet shrinkage, and the simple nonlinear noise reduction method. Buades et al. [10] introduce a non-local means algorithm and compare it to local smoothing filters. The Kalman filter is an earlier method of denoising that provides a recursive solution to discrete-data linear filtering [97]. Lalley and Nobel [48] also consider denoising in deterministic systems. Stephanedes and Chassiakos [80] illustrate the use of denoising techniques on traffic incident detection. Robertson et al. [71] introduce a least-squares estimation based on the moving horizon approach. The approach is similar to that of the moving control horizon in Model Predictive Control (MPC) [61, 68]. Zavala et al. [104] propose a fast moving horizon estimation algorithm that is based on Nonlinear Model Predictive Control (NLP) sensitivity concepts as well as background optimization. All of these studies utilize different variations on denoising to reduce the number of outliers and relatively clean up the data sets.

Jade et al. [38] combine feature selection with denoising in a kernel PCA method. They demonstrate this method on a chaotic time series and an input-output model for polymer nanocomposites. The kernel PCA method proves capable of extracting a

large number of principle components, thus allowing for the successful combining of feature selection and denoising within one approach.

Takeda et al. [84] provide an approach that incorporates regression with denoising. It provides evidence that the adaptive kernel regression performs better than state of the art methods. Baecher [5] shows that kernel regression framework allows for successful denoising. Chuang et al. [17] create a Robust Support Vector Regression (RSVR) to suppress over-fitting due to the possible presence of outliers in the SVR.

The proper individual use of regression, denoising, and feature selection methods, helps to find solutions with improved computing efficiently. We also find that the collaboration of two or more of these methods can reduce problem areas, such as outliers, while increasing the efficiency of the combinatorial methodology.

The main approach in this study is to find the contribute features on a dependent variable, using regression facts, meant for bags of data sets. Examples of such studies include protein family modeling (see [85]), stock prediction (see [54]), content-based image retrieval (see [55]), and text classification (see [3]).

Next, we describe mathematically ε-insensitive regression method and then present our formulation for regression with denoising.

4.3 Mathematical Modeling

In this section we present the base mathematical core of our method which is called regression with denoising. This method is based on ε-insensitive regression method which has been introduced by Vapnik [90]. First, we briefly introduce the traditional ε-insensitive regression methods then we continue explaining our formulation.

Suppose $\mathbf{X}$ as a set of given pattern vectors $\mathbf{x}_i \in \mathbb{R}^d$, with dependent variable values (i.e., real-valued response) $y_i \in \mathbb{R}$. The linear ε-insensitive loss function $L^{\varepsilon}(\mathbf{x}, y, f)$

can be defined as

$$L^{\varepsilon}(\mathbf{x}, y, f) = |y - f(\mathbf{x})|_{\varepsilon} = \max(0, |y - f(\mathbf{x})| - \varepsilon). \tag{4.1}$$

To achieve the ε-insensitive regression loss function, a regression function $f(\cdot)$ with at most ε deviation from y_i shall be defined in a way that, if the distance between the regression hyperplane and our real-valued response is more than ε an associated penalty of C will incur, otherwise this error will be neglected. It is desirable that the data would be in ε band of regression hyperplane which is defined by $\boldsymbol{\psi}$ and b, where $\boldsymbol{\psi}$ is the norm of the regression hyperplane (also called the weight vector) and b is the constant term called the regression independent term.

If $\mathbf{x}^*$ would be equal to $|(\langle \boldsymbol{\psi}, \mathbf{x}^* \rangle) + b|$ which is the distance between the regression hyperplane and the closest pattern vector, the solution of following quadratic programming problem is the regression hyperplane with the minimum sum of quadratic ε-insensitive losses:

$$\begin{aligned}
\min_{\boldsymbol{\psi}, b, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}} \quad & \frac{1}{2}\|\boldsymbol{\psi}\|^2 + C \sum_{i=1}^{n} \left(\xi_i + \hat{\xi}_i\right) && && \text{(4.2a)} \\
\text{subject to} \quad & (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) - y_i \leq \varepsilon + \xi_i && \forall i \in I && \text{(4.2b)} \\
& y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) \leq \varepsilon + \hat{\xi}_i && \forall i \in I && \text{(4.2c)} \\
& \xi_i, \hat{\xi}_i \geq 0 && \forall i \in I, && \text{(4.2d)}
\end{aligned}$$

where ξ_i and $\hat{\xi}_i$ are slack variables which represent errors associated with instances outside the ε boundaries and will be zero for all instances inside the insensitive band. This convex optimization problem minimizes the penalty cost to reveal the best regression fit to the model, constrained with (4.2b-4.2c) which imply that the pattern

vectors are allowed to be ε below or above the target value without penalty respectively. All pattern vectors outside the ε range are still allowed, however they incur a cost of C (see [76]).

Differentiating the Lagrangian function for (4.2) with respect to the primal variables, the dual formulation for the mentioned optimization problem can be written as

$$\begin{aligned}
\min_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\hat{\alpha}_i)(\alpha_j-\hat{\alpha}_j)\langle \mathbf{x}_i,\mathbf{x}_j\rangle \\
& +\varepsilon\sum_{i=1}^{n}(\alpha_i+\hat{\alpha}_i)+\sum_{i=1}^{n}y_i(\alpha_i-\hat{\alpha}_i) && \text{(4.3a)}\\
\text{subject to} \quad & \sum_{i=1}^{n}(\alpha_i-\hat{\alpha}_i)=0 && \text{(4.3b)}\\
& 0\le \alpha_i,\hat{\alpha}_i\le C \qquad\qquad \forall i\in I. && \text{(4.3c)}
\end{aligned}$$

From the solution $\boldsymbol{\alpha}^*$ and $\hat{\boldsymbol{\alpha}}^*$, the regression function can be written as $f(\mathbf{x}) = \sum_{i=1}^{n}(\hat{\alpha}_i^*-\alpha_i^*)\langle \mathbf{x}_i,\mathbf{x}\rangle + b^*$, where b^* is chosen such that $f(\mathbf{x}_i)-y_i=-\varepsilon$ for any i with $0<\hat{\alpha}_i^*<C$.

The dual formulation has the significant advantage of using nonlinear maps to embed the pattern vectors in a higher dimensional space in such a way that a hyperplane can perform regression for the mapped pattern vectors in the embedded space. This embedding is done via the *kernel trick*. Kernels enhance similarity measures between pattern vectors. The mapping is defined over dot product *Hilbert* spaces. This transformation is done by replacing the dot product $\langle \mathbf{x}_i\cdot\mathbf{x}_j\rangle$, with a nonlinear kernel $K(\mathbf{x}_i,\mathbf{x}_j)$.

The fundamental ε-insensitive technique for denoising can be formulated in general form by defining the input data incorporating the notion of *bags*. Let $\mathbf{x}_1,\dots\mathbf{x}_n$ be a set of patterns that are grouped into bags $\mathbf{X}_1,\dots\mathbf{X}_m$ with $\mathbf{X}_j=\{\mathbf{x}_i : i\in I_j\}$,

$I_j \subseteq \{1, \ldots, n\}$, each instance $\mathbf{x}_i$ is associated with a dependent variable value $y_i \in \mathbb{R}$. Therefore, each bag can have multiple responses. Bags create a notion of selection of the most significant instance regarding the output contribution. *Exemplary (primary) instances* issue the following statement: "*One pattern in each bag is an example of its associated response*".

Using the bag notion as explained above, regression with denoising problem reduces to selecting exactly one pattern vector (*primary instance*) from each bag such that the weighted sum of the ε-insensitive errors for the selected pattern vectors and regularization term is going to be minimized as follows:

$$\min_{\boldsymbol{\psi}, b, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}, \boldsymbol{\eta}} \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + C\sum_{i=1}^{n}(\xi_i + \hat{\xi}_i) \tag{4.4a}$$

$$\text{subject to:} \quad (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) - y_i \leq \varepsilon + \xi_i, \text{ if } \eta_i = 1 \qquad \forall i \in I \tag{4.4b}$$

$$y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i \rangle + b) \leq \varepsilon + \hat{\xi}_i, \text{ if } \eta_i = 1 \qquad \forall i \in I \tag{4.4c}$$

$$\sum_{i \in I_j} \eta_i = 1 \qquad \forall j \in J \tag{4.4d}$$

$$\eta_i \in \{0, 1\} \qquad \forall i \in I, \tag{4.4e}$$

where η_i is a binary variable denotes the selection status of an instance. M is a sufficiently large number, such that when $\eta_i = 0$, both (4.5b) and (4.5c) are always satisfied, hence the associated instance does not have any influence on the problem which yields removing this pattern vector (that can be considered noise) from the problem. Constraints (4.5b,4.5c) ensure a pattern vector is penalized if it is outside the ε-insensitive band. Finally, the constraint (4.5d) guaranties that only one pattern vector from each bag will be selected. The regression with denoising problem can be formulated as the following quadratic mixed 0–1 programming problem considering M as an arbitrarily large number:

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\hat{\boldsymbol{\xi}},\boldsymbol{\eta}} \quad \frac{1}{2}\|\boldsymbol{\psi}\|^2 + C\sum_{i=1}^{n}(\xi_i + \hat{\xi}_i) \tag{4.5a}$$

$$\text{subject to:} \quad (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) - y_i \leq \varepsilon + \xi_i + M(1-\eta_i) \qquad \forall i \in I \tag{4.5b}$$

$$y_i - (\langle \boldsymbol{\psi}, \mathbf{x}_i\rangle + b) \leq \varepsilon + \hat{\xi}_i + M(1-\eta_i) \qquad \forall i \in I \tag{4.5c}$$

$$\sum_{i\in I_j} \eta_i = 1 \qquad \forall j \in J \tag{4.5d}$$

$$\eta_i \in \{0,1\} \qquad \forall i \in I. \tag{4.5e}$$

In order to apply the *kernel trick* for regression with denoising, we used the method in 4.4 to achieve the dual problem which can be explained as follows:

$$\begin{aligned}\min_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}},\boldsymbol{\eta}} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n}(\eta_i\alpha_i - \eta_i\hat{\alpha}_i)(\eta_i\alpha_i - \eta_i\hat{\alpha}_i)K(\mathbf{x}_i,\mathbf{x}_j) \\ & + \varepsilon\sum_{i=1}^{n}(\eta_i\alpha_i + \eta_i\hat{\alpha}_i) + \sum_{i=1}^{n} y_i(\eta_i\alpha_i - \eta_i\hat{\alpha}_i)\end{aligned} \tag{4.6a}$$

$$\text{subject to:} \sum_{i=1}^{n}(\eta_i\alpha_i - \eta_i\hat{\alpha}_i) = 0 \tag{4.6b}$$

$$0 \leq \alpha_i, \hat{\alpha}_i \leq C \qquad \forall i \in I \tag{4.6c}$$

$$\sum_{i\in I_j} \eta_i = 1 \qquad \forall j \in J \tag{4.6d}$$

$$\eta_i \in \{0,1\} \qquad \forall i \in I, \tag{4.6e}$$

where α_i and $\hat{\alpha}_i$ represent Lagrangian multipliers and inner products are replaced by kernel function $K(\mathbf{x}_i,\mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle$ to perform nonlinear regression for selecting problem. New binary variables (z_i and $\hat{z}_i$) have been used to linearize the constraint (4.7b). After considering this linearization, the nonlinear regression with denoising formulation is as follows:

$$\min_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}},\boldsymbol{\eta},\mathbf{z},\hat{\mathbf{z}}} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n}(z_i-\hat{z}_i)(z_i-\hat{z}_i)K(\mathbf{x}_i,\mathbf{x}_j) + \varepsilon\sum_{i=1}^{n}(z_i+\hat{z}_i)+\sum_{i=1}^{n}y_i(z_i-\hat{z}_i) \tag{4.7a}$$

$$\text{subject to:} \sum_{i=1}^{n}(z_i-\hat{z}_i)=0 \tag{4.7b}$$

$$0 \le z_i, \hat{z}_i \le C\eta_i \qquad \forall i \in I \tag{4.7c}$$

$$z_i \le \alpha_i \qquad \forall i \in I \tag{4.7d}$$

$$\hat{z}_i \le \hat{\alpha}_i \qquad \forall i \in I \tag{4.7e}$$

$$z_i \ge \alpha_i - C(1-\eta_i) \qquad \forall i \in I \tag{4.7f}$$

$$\hat{z}_i \ge \hat{\alpha}_i - C(1-\eta_i) \qquad \forall i \in I \tag{4.7g}$$

$$0 \le \alpha_i, \hat{\alpha}_i \le C \qquad \forall i \in I \tag{4.7h}$$

$$\sum_{i\in I_j}\eta_i = 1 \qquad \forall j \in J \tag{4.7i}$$

$$\eta_i \in \{0,1\} \qquad \forall i \in I. \tag{4.7j}$$

It should be noted that this problem is $\mathcal{NP}$-hard since a special case with ambiguous labels (where bags share a common label) is proven to be $\mathcal{NP}$-hard for bag sizes of at least 3 (see [93]). Next, we will explain our solution approach for this problem.

4.4 Solution Approach

In this section, we propose an algorithm for the problem in hand. First, we explain feature selection algorithm which is used with regression and denoising model and then we describe our heuristic algorithm that is useful for large scale problems.

4.4.1 Feature Selection Algorithm

The objective of feature selection algorithm is to find the features that have a significant contribution to the response variable. The algorithm uses the proposed regression formulation as the underlying mathematical model to simultaneously remove the noise, while finding the most important features. First, we start with applying regression with denoising for all bags to have a rough idea about the potential features that are significant. Then we sort the features that do not contribute more than a pre-defined threshold in ascending order of their coefficient's[1] absolute value and choose the smallest one. We then examine to see whether this feature can be removed from our data. For this purpose we temporarily remove that feature and apply regression with denoising and check the R^2 of the model. If it is not changed significantly, we remove the selected feature from our data set and start over to find the next feature. If R^2 has changed significantly, we suspect that feature is important despite the small coefficient value. Thus we keep that feature and select another one from our sorted feature list. We continue this process until all features contribute more than the pre-defined threshold.

4.4.2 Heuristic Algorithm for Regression with Denoising

The idea of our heuristic algorithm for regression with denoising is to start with a feasible hyperplane and fine tune the orientation considering MIL restrictions. Instead of starting with a random hyperplane, we take advantage of the efficiency of ε- insensitive regressors on our problem. Therefore, the first phase of the algorithm consists of applying ε- insensitive regressors on all instances regardless of their bags. We use LIBSVM [34] since a fast regression of the data set is needed. The optimal regression hyperplane in this step $(\boldsymbol{\psi}_1, b_1)$ gives a rough idea on positioning of bags.

[1]The coefficient here refers to the weight of the orthonormal vector of the regression hyperplane.

Next, we continuously update the solution by removing instances from bags. For this purpose, we choose an instance ($\mathbf{x}_i$) which has the most error (difference of its response from its estimated value by regressor ($|\langle \boldsymbol{\psi}_1, \mathbf{x}_i \rangle + b_1 - y_i|$)) and remove it from our data. We apply ε- insensitive regression to change the orientation of our regression hyperplane. We continue this process until one instance from each bag remains. The optimal regression hyperplane of this step is $(\boldsymbol{\psi}_2, b_2)$ that supposedly gives a better representation of data. Finally, we compute current objective function value using $\|\boldsymbol{\psi}_2\|^2$ and errors associated with instances outside the ε boundaries. This hyperplane also becomes the *current* best solution.

The goal of the second phase is to improve the objective function. For this purpose we seek for instances that can be substituted with the currently selected instances in each bag. We calculate the errors ($|\langle \boldsymbol{\psi}_2, \mathbf{x}_i \rangle + b_2 - y_i|$) and choose the one which has the least error in each bag. If the new set of instances are not the same as previously selected instances, we apply ε- insensitive regression to come up with the new orientation. If the objective value is less than the current best objective, candidate hyperplane will be updated. The search will continue until no improvement is possible and the final best solution is the heuristic solution of the problem $(\boldsymbol{\psi}_3, b_3)$. The pseudocode is presented in Algorithm 2.

For traditional ε-insensitive regressor, we use an open source software IBSVM [34]. Therefore, the nonlinear extension of Algorithm 2 only requires calling LIBSVM software using nonlinear regression option. Next, we report computational performance for the proposed algorithm.

4.5 Computational Results

Natural gas, a mixture of hydrocarbon and non-hydrocarbon gases, is one of the abundant sources of energy in United States. In United States, because of recent

Algorithm 2 Two-Phase Heuristic Algorithm (Linear Regression)
INPUT: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$**,** J**,** I**,** C
OUTPUT: $\boldsymbol{\psi}_{best}, b_{best}, Objective_{best}$

{**PHASE I**}
$I_{active} \leftarrow I$
$J_{active} \leftarrow J$
$N_j \leftarrow |I_j|$
$\boldsymbol{\psi}_1, b_1 \leftarrow$ traditional ε-insensitive regressors for I_{active}
while $\sum_j N_j > |J|$ **do**
  $i_{selected} \leftarrow arg\max_{i \in I_j, j \in J_{active}} |\langle \boldsymbol{\psi}_1, x_i \rangle + b_1 - y_i|$
  $j_{selected} = j : i_{selected} \in I_j$
  $I_{active} \leftarrow I_{active} \setminus i_{selected}$
  $N_{j_{selected}} \leftarrow N_{j_{selected}} - 1$
  if $N_j = 1$ **then**
    $J_{active} \setminus j_{selected}$
  end if
end while
$\boldsymbol{\psi}_2, b_2 \leftarrow$ traditional ε-insensitive regressor for I_{active}
$Objective_{best} \leftarrow \frac{1}{2}\|\boldsymbol{\psi}_2\|^2$
for all $i \in I$ **do**
  $Objective_{best} \leftarrow Objective_{best} + C \times (\langle \boldsymbol{\psi}_2, x_i \rangle + b_2 - y_i|)$
end for

{**PHASE II**}
$I_{min} \leftarrow \emptyset$
$\boldsymbol{\psi}_{best} \leftarrow \boldsymbol{\psi}_2$
$b_{best} \leftarrow b_2$
status←false
while status=false **do**
  Empty I_{min}
  for all $j \in J$ **do**
    $I_{min} \cup arc\min_{i \in I_j} (|\langle \boldsymbol{\psi}_{best}, x_i \rangle + b_{best} - y_i|)$
  end for
  if $I_{min} \neq I_{active}$ **then**
    $I_{active} \leftarrow I_{min}$
    $\boldsymbol{\psi}_3, b_3 \leftarrow$ traditional ε-insensitive regressors for I_{active}
    $Objective_{new} \leftarrow \frac{1}{2}\|\boldsymbol{\psi}_3\|^2$
    for all $i \in I$ **do**
      $Objective_{new} \leftarrow Objective_{new} + C \times (\langle \boldsymbol{\psi}_3, x_i \rangle + b_3 - y_i|)$
    end for
    if $Objective_{new} < Objective_{best}$ **then**
      $\boldsymbol{\psi}_{best} \leftarrow \boldsymbol{\psi}_3$
      $b_{best} \leftarrow b_3$
      status← true
    end if
  end if
end while

technological advances and plentiful reservoirs, shale gas usage is soaring. As a result, finding the features affecting gas prices has great significance to industries and produce potential economic benefits. For identifying variables that potentially affect Henry Hub natural gas prices, we conducted a comprehensive literature review and a survey from technical experts in the fields of energy, scheduling, and planning. As a result, following variables are recognized: electricity price, storage, Dow Jones index, NYSE index, Dow Jones coal index, U.S. natural gas pipeline imports, U.S. LNG imports,

U.S. natural gas consumption, U.S. natural gas gross withdrawals, U.S. natural gas marketed production, renewable energy production, renewable energy consumption, shale gas production, weather (temperature), and WTI oil spot price. The historical data for all of these variables is available concurrently for 2003 to 2009 period and is gathered from different resources, including U.S. Energy Information Administration (EIA) [88] and [96]. The shale gas production (MMcf/d) information is generated through the implementation of a digitizer software on the data published in [82].

The data gathered from these various resources is not in the same time format and uniform. Some of these data are available in days, and some in weeks and months; therefore, considering the low variance within each weak, we calculate the weekly average for daily data, including Dow Jones index data, NYSE index, Dow Jones Coal index and natural gas Henry Hub historical prices. Next, we consider taking the monthly averages but for certain weeks some artificial rapid changes are observed in a number of variables including the Henry Hub historical price. These changes is considered as noises and should be removed for more accurate forecast model. In order to satisfy uniformity, weekly data is assumed to constitute each data instance. The variables whose data was available monthly and are assumed to be constant through the weeks of the month are: electricity price, U.S. natural gas pipeline imports, U.S. LNG imports, U.S. natural gas total consumption, U.S. natural gas gross withdrawals, U.S. natural gas marketed production, total renewable energy production, total renewable energy consumption and WTI oil spot price. Our goal is to *simultaneously identify and omit artificial changes* in independent variables (potential features affecting price) and response (Henry Hub natural gas price) and *diminish the impact of monthly to weekly conversion.*

For validation purposes, we use the Root Mean Squared Logarithmic Error (RMSLE) to measure the accuracy of our algorithm given as

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1) - \log(a_i+1))^2}, \tag{4.8}$$

where the natural logarithms are considered. In this equation, p_i and a_i are the predicted and actual values for data instance i, respectively.

The independent variable values to be plugged in for prediction can be forecasted using time series forecasting (e.g., moving average) or an expert opinion in cases such as weather. At this point, in order to assess the performance of regression with denoising, we keep prediction of independent variables as a secondary objective and assume future values are known for these variables.

Figure 4.1 shows the 3-month-ahead forecasts obtained using simple regression, regression with denoising, and 12 months weighted moving average since it provides best forecasts among time series methods.

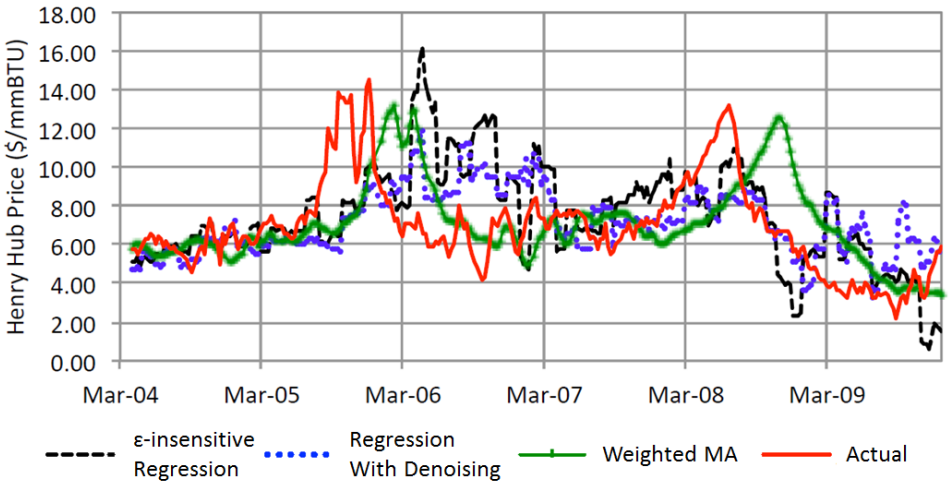

Figure 4.1: 3-month-ahead Henry Hub price forecasting with ε-insensitive regression, regression with denoising and 12 months weighted moving average.

To assess the consistency of these results, each method is applied to forecast

different time periods ahead: 2 months, 3 months, 4 months, 5 months, 6 months, 7 months, and 12 months. The final predictions are assessed by their RMSLE value. Those values are plotted in Figure 4.2.

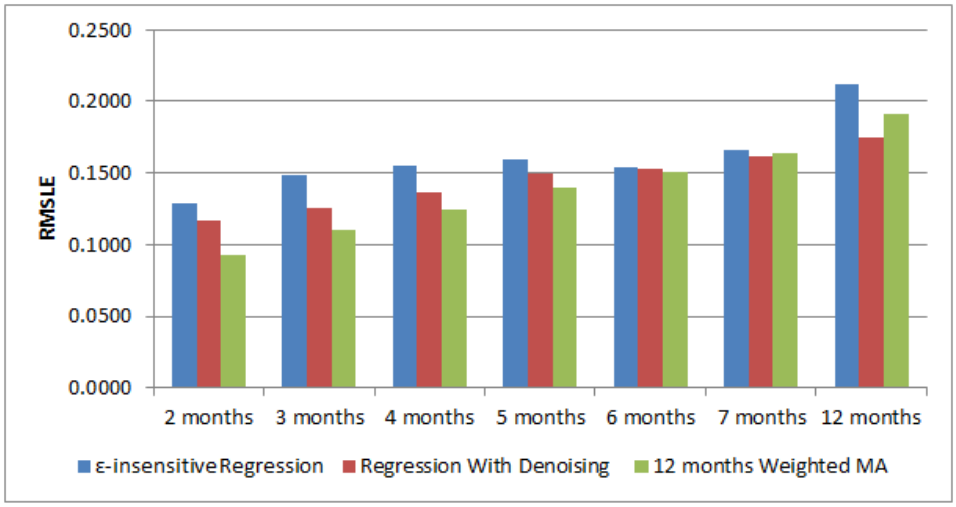

Figure 4.2: Weekly RMSLE for each method when predicting different time periods.

Regression with denoising consistently provides lower RMSLE than traditional regression. Even though weighted moving average has the lowest RMSLE for 2 months ahead to 6 months ahead, it shows a constant increasing trend that is evident when predicting for a period of 12 months. In fact, this is due to MA's infamous lagging from behind weakness in forecasts. Next, we analyze the performance of the same methods using their minimum errors within each month and present the RMSLE based on these exemplary weeks in Figure 4.3. As expected, this boosts the performance of regression with denoising further as this evaluation process also assumes potential existence of outlier observations.

Finally, we compare our 3-month-ahead regression with denoising forecasts with the U.S. Energy Information Administration (EIA) Short Term Energy Outlook (STEO) forecast for the Henry Hub prices. The information with the STEO historical forecast is public and available in [88]. The STEO projections are updated every month and include forecast for the next 12 months. The outlooks are available

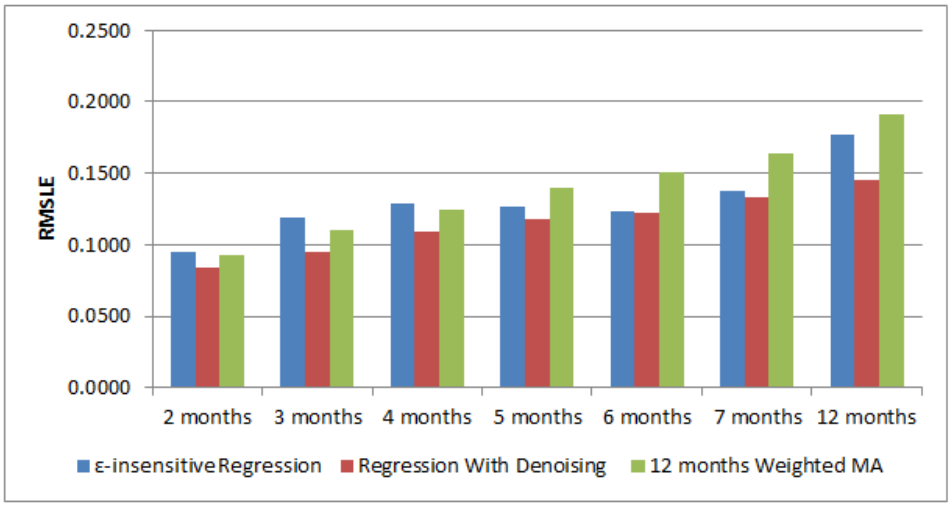

Figure 4.3: Monthly minimum RMSLE for each method when predicting different time periods.

in excel data file since 2004 and in full PDF report since 1983. Note that there are no technical details in how the projections are calculated. The STEO Henry Hub price forecast is available per month. Comparisons with regression with denoising results using actual and forecasted independent variables and the STEO forecast are conducted. Figure 4.4 shows the STEO performance for a 3-month-ahead forecast along with the outputs of the aforementioned methods assuming the independent variables are known 3 months in advance.

Having shown the success of regression with denoising especially in a longer forecasting horizon, next we present results for traditional regression and regression with denoising using the predictions of independent variables. Clearly, this is a more realistic setting for forecasting natural gas prices. Figure 4.5 shows the STEO performance, 12 months weighted moving average on natural gas prices and the performances of the traditional regression and the regression with denoising using forecasted independent variable data. The RMSLE values obtained by four methods are summarized in Table 4.1.

Regression with denoising (RMSLE 0.0990) has a slightly more accurate forecast

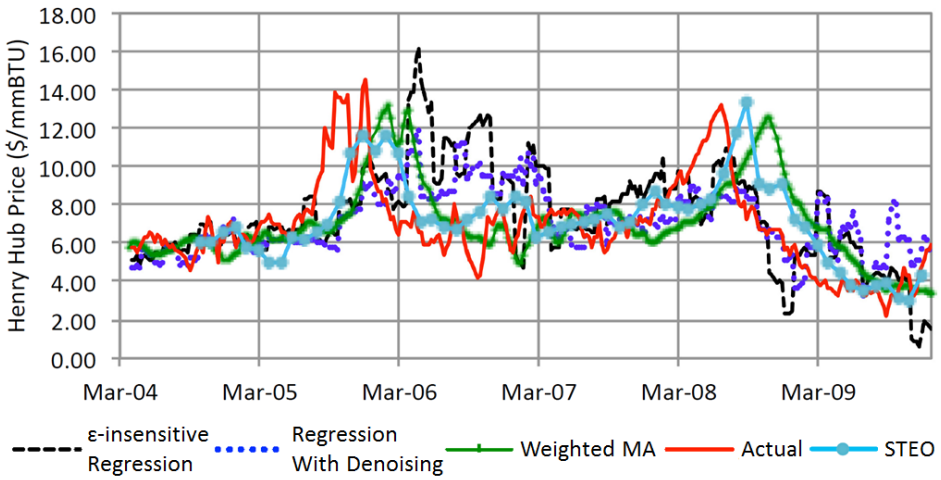

Figure 4.4: 3-month-ahead forecasts: traditional regression, regression with denoising, STEO, and 12 months weighted moving average assuming independent variables data is known.

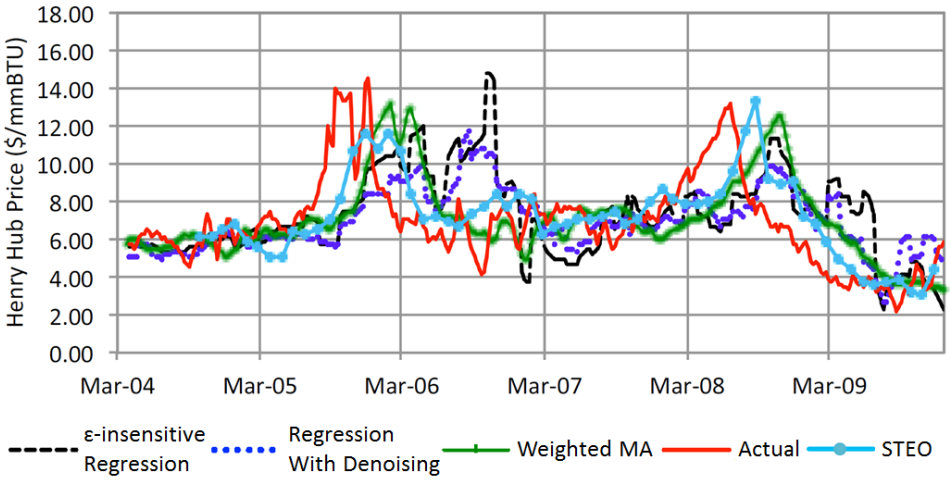

Figure 4.5: 3-month-ahead forecasts: traditional regression, regression with denoising, STEO, and 12 months weighted moving average using MA forecasted independent variable data.

than the STEO method (RMSLE of 0.0994) for a 3 months forecast. When the independent variables are assumed known, error measure for regression with denoising is decreased further. Aside from a small difference in accuracy, note that one of the greatest advantages of a causal model is the ability to clearly highlight factors that affect natural gas price. Having validated our approach with natural gas price data,

Table 4.1: Comparison of RMSLE values for traditional regression, regression with denoising, STEO, and 12 months weighted moving average 3-month-ahead forecasts.

Independent Variables	ε – insensitive Regression	Regression with Denoising	12 months Weighted MA	STEO
Known	11.88%	**9.54%**	10.98%	9.94%
Forecasted	12.19%	**9.91%**		

next we present our feature selection results.

Feature selection procedure consists of iterative applications of presented regression methods on the data. During this procedure, we ensure that R^2 values do not decrease drastically for the training set and we try to minimize the number of selected features. Significance of features on natural gas prices are observed after applying regression with denoising with this feature selection routine and results are shown in Figures 4.6 and 4.7. These results also show the expected recent disparity between the price of oil (WTI) and natural gas, yet there still exists some correlation especially in the years of 2005 and 2008 which are known to be ill-posed in terms of oil prices. Aside from the fuzzy behaviors in 2005 and 2008, Figure 4.7 clearly shows the correlation between independent variables and natural gas price. For instance, before 2005, U.S. LNG imports, U.S. natural gas consumption, U.S. natural gas gross withdrawals, U.S. natural gas marketed production, and weather seem to affect natural gas price consistently. Likewise, between 2005 and 2008 we can observe the consistent effect of electricity price, U.S. natural gas pipeline imports, weather as well as *renewable energy production* and *renewable energy consumption* using Figure 4.7. On the other hand, a similar consistent insight cannot be derived from Figure 4.6.

4.6 Summary

In this chapter we present a novel method to improve feature selection with simultaneously removing the noise while performing the learning process. Linear and

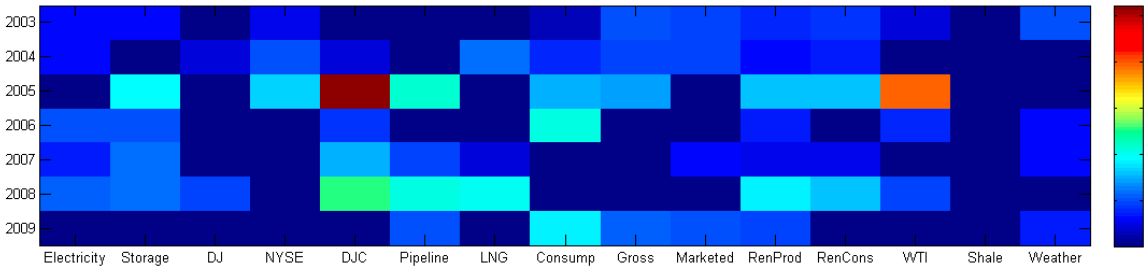

Figure 4.6: Heatmap of features' coefficients (absolute values) in the regression with traditional regressor.

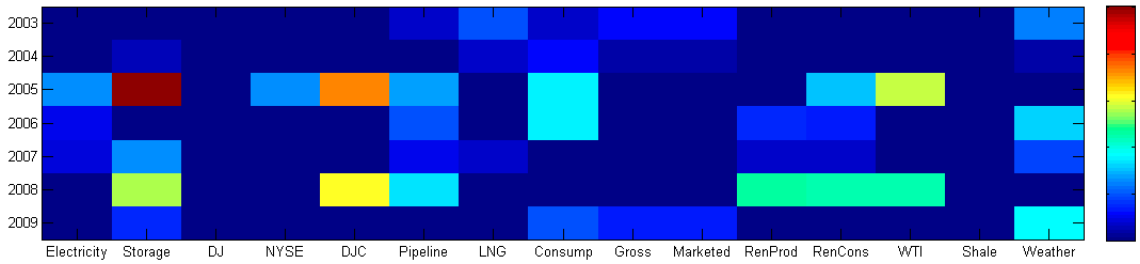

Figure 4.7: Heatmap of features' coefficients (absolute values) in the regression with denoising regressor.

nonlinear formulations are developed to formulate our regression with denoising problem. In order to validate our method we analyze the relationship between potential variables that have an impact on the Henry Hub natural gas price and the historical natural gas price behavior. We compared our method with moving average and traditional regression approaches. The results show our approach has better performance in terms of accuracy of predictions.

Chapter 5 Conclusions and Future Work

In this dissertation, we studied three data mining and optimization approaches that can be used for sustainable energy applications. In this chapter we summarize the research that has been completed for each of these studies. Next, we discuss problems and solution methods that can be explored in the future.

5.1 Concluding Remarks

In Chapter 2, we proposed a robust approach for Multiple Instance Learning based on Support Vector Machines. We used three different loss functions and showed that hard margin loss classifiers provide better generalization performance for multiple instance data. Three nonlinear integer programs, two constraint programs, and a nonlinear classifier are developed in this study. The results show that these formulations can solve medium size problems to optimality in reasonable time. Since the problem is $\mathcal{NP}$-hard, we developed a three-phase heuristic algorithm that can handle large problem instances within seconds. We extended traditional cross validation approaches to consider bags for multiple instance data and compared our heuristic with conventional hinge loss based Support Vector Machines (SVMs). The leave one bag out cross validation (LOBOCV) result shows our heuristic provides higher accuracy for multiple instance data in significantly less time. We implemented our method on wind farm site locating problem and showed promising results.

Offshore wind farm layout optimization framework has been studied in Chapter 3. We presented a mathematical model that would minimize the cost of wind energy by increasing the number of turbines to generate more power while considering loss generated by wake effects. By using two linearization techniques we came up with a general formulation that is useful for a wide-variety of wind scenarios and power

curves discussed in the literature. Mosseti's problem instances have been used to validate our method. We find the best layout in each problem instance by increasing the number of turbines located and maximizing the generated power.

In Chapter 4, we developed a novel optimization algorithm that is capable of simultaneously removing noise and performing regression. We used this algorithm to analyze historical information on variables that potentially have an impact on the response variable. We considered a framework that consists of sets of instances that are tied in some way and the number of outliers in each set is bounded from above. Two linear and nonlinear regressors are developed to address this problem. We used natural gas pricing data to validate our approach and developed an efficient price prediction tool. Traditional ε-insensitive regression and Weighted Moving Average (WMA) methods are used for benchmarking purposes. The Root Mean Squared Logarithmic Error (RMSLE) results show our method outperforms both WMA and traditional regression in terms of accuracy of predictions.

5.2 Future Work

This section includes the overview of future research and possible progressions in our problems. Each chapter in this dissertation can be considered as a starting point for further investigation. We introduce several topics that can be chosen for further advanced research.

In Chapter 2, we observed that ramp loss classifiers are slow in practice. Alternative formulations can be developed and problem complexity can be studied for ramp loss SVM for conventional and multiple instance data. Another important future study may be a comparison of approaches highlighted in Section 2.2.2 using a fair cross validation scheme (e.g., leave one bag out), instead of random validation schemes that generate varying results in different runs.

In Chapter 3, we recommend considering different wake effect models and power curves. Jensen's model is claimed to have imperfections in modeling the wake behind the turbines, therefore this study may help understand the sensitivity of our model to the functions employed.

In Chapter 4, results indicate the prediction of the independent variables can be further enhanced that would ultimately improve the prediction of the response variable (natural gas price). Although the performance of the regression with denoising is not severely affected using forecasted data, some of the independent variables we consider can be forecasted more accurately for a better overall performance. The periods of peaks and valleys that are not artificial, cause a serious decrease in performance of our method. When isolated periods of no particular trend are considered, regression with denoising achieves less than 3% RMSLE. Therefore, these trend changes and peaks can be further analyzed to understand and identify conditions and contribute to the learning procedure.

References

[1] 4C Offshore Co., 2012. URL `http://www.4coffshore.com/`.

[2] J. F. Ainslie. Calculating the flowfield in the wake of wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 27(1):213–224, 1988.

[3] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *Advances in Neural Information Processing Systems*, pages 577–584, 2003.

[4] European Wind Energy Association. *Wind Energy - The Facts: A Guide to the Technology, Economics and Future of Wind Power*. Routledge, 2009.

[5] M. Baecher. Locally weighted least squares regression for image denoising, reconstruction and up-sampling. 2009.

[6] R. J. Barthelmie, G. C. Larsen, S. T. Frandsen, L. Folkerts, K. Rados, S. C. Pryor, B. Lange, and G. Schepers. Comparison of wake model simulations with offshore wind turbine wake profiles measured by sodar. *Journal of Atmospheric and Oceanic Technology*, 23(7):888–901, 2006.

[7] C. Bergeron, F. Cheriet, J. Ronsky, R. Zernicke, and H. Labelle. Prediction of anterior scoliotic spinal curve from trunk surface using support vector regression. *Engineering Applications of Artificial Intelligence*, 18(8):973–983, 2005.

[8] J. P. Brooks. Support vector machines with the ramp loss and the hard margin loss. *Operations Research*, 59(2):467–479, 2011.

[9] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene

expression data by using support vector machines. In *National Academy of Sciences of the United States of America*, volume 97, pages 262–267, 2000.

[10] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[11] H. Byun and S. W. Lee. Applications of support vector machines for pattern recognition: a survey. *Pattern Recognition with Support Vector Machines*, pages 571–591, 2002.

[12] E. F. Camacho, T. Samad, M. Garcia-Sanz, and I. Hiskens. Control for renewable energy and smart grids. *The Impact of Control Technology, Control Systems Society*, pages 69–88, 2011.

[13] Cape Wind. America's first offshore wind farm on nantucket sound, 2012. URL `http://www.capewind.org/index.php`.

[14] C. Chen and O. L. Mangasarian. Hybrid misclassification minimization. *Advances in Computational Mathematics*, 5:127–136, 1996.

[15] Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, 2004.

[16] K. Y. Choy and C. W. Chan. Modelling of river discharges and rainfall using radial basis function networks based on support vector regression. *International Journal of Systems Science*, 34(14–15):763–773, 2003.

[17] C. C. Chuang, S. F. Su, J. T. Jeng, and C. C. Hsiao. Robust support vector regression networks for function approximation with outliers. *Neural Networks, IEEE Transactions on*, 13(6):1322–1330, 2002.

[18] A. Crespo, J. Hernandez, and S. Frandsen. Survey of modelling methods for wind turbine wakes and wind farms. *Wind Energy*, 2(1):1–24, 1999.

[19] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, UK, 2000.

[20] D. Danielson. A banner year for the U.S. wind industry, August 14 2012. URL `http://energy.gov/articles/banner-year-us-wind-industry`.

[21] M. Desholm and J. Kahlert. Avian collision risk at an offshore wind farm. *Biology Letters*, 1(3):296–298, 2005.

[22] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2): 31–71, 1997.

[23] D. R. Dooly, Q. Zhang, S. A. Goldman, and R. A. Amar. Multiple-instance learning of real-valued data. *Journal of Machine Learning Research*, 3:651–678, 2002.

[24] C. N. Elkinton, J. F. Manwell, and J. G. McGowan. Offshore wind farm layout optimization (OWFLO) project: Preliminary results. *University of Massachusetts*, 2006.

[25] F. J. Ferri, P. Pudil, M. Hatef, and J. Kittler. Comparative study of techniques for large-scale feature selection. *Machine Intelligence and Pattern Recognition*, 16:403–403, 1994.

[26] I. K. Fodor. A survey of dimension reduction techniques. *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory*, 9:1–18, 2002.

[27] A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

[28] P. Fuglsang and K. Thomsen. *Cost optimization of wind turbines for large-scale offshore wind farms*. Forskningscenter Risø, Roskilde, 1998.

[29] P. V. Gehler and O. Chapelle. Deterministic annealing for multiple-instance learning. In *International Conference on Artificial Intelligence and Statistics*, 2007.

[30] Y. Grandvalet and S. Canu. Adaptive scaling for feature selection in svms. *Advances in neural information processing systems*, 15:553–560, 2002.

[31] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[32] G. L. G. Hassan. Wake effects within and between large wind projects: The challenge of scale, density and neighbours-onshore and offshore. 2010.

[33] W. C. Hong and P. F. Pai. Potential assessment of the support vector regression technique in rainfall forecasting. *Water Resources Management*, 21(2):495–513, 2007.

[34] C. W. Hsu, C. C. Chang, and C. J. Lin. A practical guide to support vector classification. http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf, 2004.

[35] Z. Huang, H. Chen, C. J. Hsu, W. H. Chen, and S. Wu. Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision Support Systems*, 37(4):543–558, 2004.

[36] K. Hyunsoo, Z. X. Jeff, M. C. Herbert, and P. Haesun. A three-stage framework for gene expression data analysis by l1-norm support vector regression.

International Journal of Bioinformatics Research and Applications, 1(1):51–62, 2005.

[37] IBM ILOG CPLEX Optimization Studio. *12.0 User's Manual*, 2011.

[38] A. M. Jade, B. Srikanth, V. K. Jayaraman, B. D. Kulkarni, J. P. Jog, and L. Priya. Feature extraction and denoising using kernel pca. *Chemical Engineering Science*, 58(19):4441–4448, 2003.

[39] N. O. Jensen. *A note on wind generator interaction.* 1983.

[40] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142, 1998.

[41] I. Katic, J. Højstrup, and N. O. Jensen. A simple model for cluster efficiency. 1986.

[42] T. Kohler and D. Lorenz. A comparison of denoising methods for one dimensional time series. *Bremen, Germany, University of Bremen*, 131, 2005.

[43] D. Koller and M. Sahami. Toward optimal feature selection. 1996.

[44] O. E. Kundakcioglu, O. Seref, and P. M. Pardalos. Multiple instance learning via margin maximization. *Applied Numerical Mathematics*, 60(4):358–369, 2010.

[45] A. Kusiak and Z. Song. Design of wind farm layout for maximum wind energy capture. *Renewable Energy*, 35(3):685–694, 2010.

[46] M. A. Lackner and C. N. Elkinton. An analytical framework for offshore wind farm layout optimization. *Wind Engineering*, 31(1):17–31, 2007.

[47] T. N. Lal, M. Schroder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Scholkopf. Support vector channel selection in BCI. *IEEE Transactions on Biomedical Engineering*, 51(6):1003–1010, 2004.

[48] S. P. Lalley and A. B. Nobel. Denoising deterministic time series. *arXiv preprint nlin/0604052*, 2006.

[49] F. Lauer and G. Bloch. Incorporating prior knowledge in support vector regression. *Machine Learning*, 70:89–118, 2008.

[50] Y. K. Li, P. L. Yang, Y. J. Jian, S. M. Ren, and H. X. Zhao. Application of support vector regression method in predicting soil erosion intensity of small watershed in the insensitive erosion areas. *Journal of Beijing Forestry University*, 29(3):93–98, 2007.

[51] P. Mahat, W. Ongsakul, and N. Mithulananthan. Optimal placement of wind turbine DG in primary distribution systems for real loss reduction. 2006.

[52] O. L. Mangasarian and E. W. Wild. Multiple instance classification via successive linear programming. *Journal of Optimization Theory and Applications*, 137(3):555–568, 2008.

[53] J. F. Manwell, C. N. Elkinton, A. L. Rogers, and J. G. McGowan. Review of design conditions applicable to offshore wind energy systems in the United States. *Renewable and Sustainable Energy Reviews*, 11(2):210–234, 2007.

[54] O. Maron. Learning from ambiguity. Technical report, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 1998. ftp://publications.ai.mit.edu/ai-publications/pdf/AITR-1639.pdf.

[55] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 341–349, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8.

[56] L. Mason, P. Bartlett, and J. Baxter. Improved generalization through explicit optimization of margins. *Machine Learning*, 38(3):243–255, 2000.

[57] MATLAB. The language of technical computing, 2011. URL `www.mathworks.com/products/matlab`.

[58] G. N. G. Molina, T. Ebrahimi, and J. M. Vesin. Joint time-frequency-space classification of eeg in a brain-computer interface application. *EURASIP Journal on Applied Signal Processing*, 2003:713–729, 2003.

[59] G. Mosetti, C. Poloni, and B. Diviacco. Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. *Journal of Wind Engineering and Industrial Aerodynamics*, 51(1):105–116, 1994.

[60] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado. Machine learning methods for predicting failures in hard drives: a multiple-instance application. *Journal of Machine Learning Research*, 6(1):783–816, 2006.

[61] K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993.

[62] M. Nandigam and S. K. Dhali. Optimal design of an offshore wind farm layout. In *International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, pages 1470–1474. IEEE, 2008.

[63] W. S. Noble. *Kernel methods in computational biology*, chapter Support vector machine applications in computational biology, pages 71–92. MIT press, 2004.

[64] C. Orsenigo and C. Vercellis. Multivariate classification trees based on minimum features discrete support vector machines. *IMA Journal of Management Mathematics*, 14:221–234, 2003.

[65] P. F. Pai and W. C. Hong. A recurrent support vector regression model in rainfall forecasting. *Hydrological Processes*, 21(6):819–827, 2007.

[66] M. H. Poursaeidi and O. E. Kundakcioglu. Robust support vector machines for multiple instance learning. *Annals of Operations Research*, DOI 10.1007/s10479-012-1241-z, forthcoming.

[67] B. P. Rašuo and A. Č. Bengin. Optimization of wind farm layout. *FME Transactions*, 38(3):107–114, 2010.

[68] J. B. Rawlings, E. S. Meadows, and K. R. Muske. Nonlinear model predictive control: A tutorial and survey. In *Preprints IFAC Symposium ADCHEM, Kyoto, Japan*, pages 203–214, 1994.

[69] D. J. Renkema. Validation of wind turbine wake models. 2007.

[70] R. A. Rivas, J. Clausen, K. S. Hansen, and L. E. Jensen. Solving the turbine positioning problem for large offshore wind farms by simulated annealing. *Wind Engineering*, 33(3):287–297, 2009.

[71] D. G. Robertson, J. H. Lee, and J. B. Rawlings. A moving horizon-based approach for least-squares estimation. *AIChE Journal*, 42(8):2209–2224, 2004.

[72] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[73] Y. Saeys, T. Abeel, and Y. Van de Peer. Robust feature selection using ensemble feature selection techniques. *Machine Learning and Knowledge Discovery in Databases*, pages 313–325, 2008.

[74] N. A. Sakhanenko and G. F. Luger. Shock physics data reconstruction using support vector regression. *International Journal of Modern Physics*, 17(9): 1313–1325, 2006.

[75] M. Samorani. The wind farm layout optimization problem. *Leeds School of Business Research Paper Series, University of Colorado at Boulder*, 2010.

[76] B. Schölkopf and A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[77] J. Serrano Gonzalez, M. Burgos Payan, and J. M. Riquelme Santos. An improved evolutive algorithm for large offshore wind farm optimum turbines layout. In *IEEE Trondheim PowerTech*, pages 1–6. IEEE, 2011.

[78] X. Shen, G. C. Tseng, X. Zhang, and W. H. Wong. On ψ-learning. *Journal of the American Statistical Association*, 98(463):724–734, 2003.

[79] I. Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, 18:768–791, 2002.

[80] Y. J. Stephanedes and A. P. Chassiakos. Application of filtering techniques for incident detection. *Journal of Transportation Engineering*, 119(1):13–26, 1993.

[81] Y. F. Sun, Y. C. Liang, C. G. Wu, X. W. Yang, H. P. Lee, and W. Z. Lin. Estimate of error bounds in the improved support vector regression. *Progress in Natural Science*, 14(4):362–364, 2004.

[82] R. P. Sutton, S. A. Cox, and R. D. Barree. Shale gas plays : A performance perspective. In *Tight Gas Completions Conference*, number November, pages 2–3, 2010.

[83] C. Szafron. Offshore windfarm layout optimization. In *9th International Conference on Environment and Electrical Engineering*, pages 542–545. IEEE, 2010.

[84] H. Takeda, S. Farsiu, and P. Milanfar. Image denoising by adaptive kernel regression. In *Proceedings of the 39th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA*, pages 1660–1665, 2005.

[85] Q. Tao, S. Scott, N. V. Vinodchandran, and T. T. Osugi. SVM-based generalized multiple-instance learning via approximate box counting. In *Twenty-First International Conference on Machine Learning*, page 101. ACM, 2004.

[86] T. B. Trafalis and R. C. Gilbert. Robust classification and regression using support vector machines. *European Journal of Operational Research*, 173(3): 893–909, 2006.

[87] T. B. Trafalis and H. Ince. Support vector machine for regression and applications to financial forecasting. In *IEEE-INNS-ENNS International Joint Conference on Neural Networks*, volume 6, pages 348–353. IEEE, 2000.

[88] U.S. Energy Information Administration (EIA). *Annual Energy Review 2011*, 2012. URL `http://www.eia.gov/`.

[89] U.S. Offshore Wind Energy Collaborative. U.S. Offshore Wind Energy: A Path Forward. Technical report, Available at: `http://www.usowc.org`, 2009.

[90] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[91] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[92] L. J. Vermeer, J. N. Sørensen, and A. Crespo. Wind turbine wake aerodynamics. *Progress in Aerospace Sciences*, 39(6):467–510, 2003.

[93] A. Viacaba, M. H. Poursaeidi, and O. E. Kundakcioglu. Natural gas price forecasting via selective support vector regression. In *IIE Annual Conference (ISERC)*, 2012.

[94] P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. *Advances in Neural Information Processing Systems*, 18, 2006.

[95] L. Wang, H. Ji, and J. Li. Training robust support vector machine with smooth ramp loss in the primal space. *Journal of the American Statistical Association*, 71:3020–3025, 2008.

[96] Weather Underground. Weather history, January 2011. URL `http://www.wunderground.com/history`.

[97] G. Welch and G. Bishop. An introduction to the kalman filter, 1995.

[98] Wind and Water Power Program: Wind Powering America. U.S. installed wind capacity and wind project locations, 3 May 2010. URL `http://www.windpoweringamerica.gov`.

[99] Z L. Wu, C. H. Li, J. K. Y. Ng, and K. R. P. H. Leung. Location estimation via support vector regression. *IEEE Transactions on Mobile Computing*, 6(3): 311–321, 2007.

[100] X. S. Xie, W. T. Liu, and B. Y. Tang. Space based estimation of moisture transport in marine atmosphere using support vector regression. *Remote Sensing of Environment*, 112(4):1846–1855, 2008.

[101] L. Xu, K. Crammer, and D. Schurmans. Robust support vector machine training via convex outlier ablation. In *21st National Conference on Artificial Intelligence*, 2006.

[102] K. Yamamoto, F. Asano, T. Yamada, and N. Kitawaki. Detection of overlapping speech in meetings using support vector machines and support vector regression. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E89A(8):2158–2165, 2006.

[103] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text

categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420. Morgan Kaufmann Publishers Inc., 1997.

[104] V. M. Zavala, C. D. Laird, and L. T. Biegler. A fast computational framework for large-scale moving horizon estimation. In *Proceedings of 8th International Symposium on Dynamics and Control of Process Systems, Cancun, Mexico*, 2007.

[105] Q. Zhang and S. A. Goldman. EM-DD: An improved multiple-instance learning technique. *Advances in Neural Information Processing Systems*, 2:1073–1080, 2002.

[106] Q. Zhang, S. A. Goldman, W. Yu, and J. E. Fritts. Content-based image retrieval using multiple-instance learning. In *19th International Conference on Machine Learning*, pages 682–689. Citeseer, 2002.