A Distributed, Model-Based Approach to Cyber-security: Application to Wind Farms

A Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Master of Science

by

Nathan James Trantham

May

2013

APPROVAL SHEET

The thesis

is submitted in partial fulfillment of the requirements

for the degree of

Master of Science

AUTHOR

The thesis has been read and approved by the examining committee:

Alfredo Garcia

Advisor

Barry Horowitz

Stephen Patek

Accepted for the School of Engineering and Applied Science:

James H. Aylor

Dean, School of Engineering and Applied Science

May

2013

Abstract

In the U.S. Department of Energy's report, '20% Wind Energy by 2030', mitigating the risk associated with owning and operating wind turbines was identified as one of wind energy's key challenges to overcome in order to promote industry growth [1]. Similar to other large power infrastructure systems, one of the most important aspects of risk associated with wind power is its vulnerability to cyber-attacks [2]. As wind power becomes increasingly integrated with the smart-grid, cyber-attacks pose a threat not only for the immediate physical damage they may cause to turbines, but also threaten to cause serious economic damage from power blackouts due to instability in the grid [2]. This paper presents a distributed, model-based intrusion detection system (IDS) algorithm that has the ability to identify the presence of certain parameter manipulating cyber-attacks within a wind farm. The algorithm draws upon existing IDS schemes such as reputation scoring and collaborative nodes, but is unique in that leverages application-layer insight gained from understanding the interaction between wind speed dynamics within a wind farm and wind turbine supervisory control. Properties from Denmark's Horns Rev wind farm were used to help develop a credible simulation environment for the algorithm's testing.

Acknowledgments

I would like to express my gratitude for the individuals who have helped me complete this thesis. In particular, I want to thank Dr. Alfredo Garcia for his guidance and support throughout the entirety of the project. Dr. Garcia provided the insight necessary for the implementation of a reputation based algorithm – a critical part of the solution. I would also like to express thanks to Dr. Barry Horowitz and Dr. Steve Patek for their participation in my committee and the insights they have provided for my project. Additionally, I would like to thank members of UVa’s WiCAT research group for their input throughout the course of my project. Lastly, I want to thank my friends and family for their support.

Contents

1. Introduction

In the U.S. Department of Energy's 2006 report, '20% Wind Energy by 2030', mitigating the risk associated with owning and operating wind turbines was identified as one of wind energy's key challenges to overcome in order to promote industry growth [1]. One of the most critical aspects of risk associated with wind power is its vulnerability to cyber-attacks [2]. Authors in [3] have demonstrated how cyber-attacks that manipulate wind turbine control parameters can cause physical damage to critical turbine components. As wind power becomes increasingly integrated with the smart-grid, cyber-attacks pose a threat not only for the immediate physical damage they may cause to turbines, but also threaten to cause serious economic damage from power blackouts due to instability in the grid [4].

An important first step in mitigating the risk of cyber-attacks is the ability to identify the presence of a cyber-attack [2]. Unfortunately, as the complexity of cyber-attacks increases, so too does the challenge of detecting them [2]. To detect cyber-attacks, some modern wind farms employ the use of intrusion detection systems (IDS). These systems typically contain algorithms that run on a centralized computer where they analyze streams of pertinent data collected from the wind farm for issues.

Similar to most critical, large-scale energy infrastructure, wind power is reliant on a Supervisory Control and Data Acquisition (SCADA) system for communication and control over its resources [5]. The SCADA system is the collection of hardware and software that allows a centralized operator to bi-directionally communicate with the system components (e.g. wind turbine supervisory controllers) [5]. It is the SCADA system that provides the IDS the data it needs for analysis. While the SCADA system provides a convenient source for information, the IDS's reliance on SCADA data poses a significant issue. The SCADA system itself has been identified as a particularly vulnerable component to cyber-attacks due to its centralized nature and important functions [4]. Malicious control over the SCADA system could result in two potentially simultaneous situations. First, an attacker with control over the SCADA system could cause grid load loss, stability violations, turbine equipment damage, and economic loss [2]. Second, a corrupted SCADA system may report deceitful or fake data back to the intrusion detection system.

The first situation has negative consequences concerning the effects of a cyber-attack, but the second situation has important implications for the intrusion detection system. With the IDS's reliance on SCADA data, a compromised SCADA system may corrupt data in such a way that it renders the IDS inept to detect a cyber-attack. This type of attack, often denoted as a man-in-the-middle attack, attempts to obfuscate the condition of the system in hopes of deceiving the intrusion detection systems. These types of attacks are expected to grow in number and complexity and pose a significant concern to cyber-physical systems such as the smart-power grid [2].

While researchers have proposed various cyber-security solutions for SCADA systems, the risk of cyber-attacks to wind farms extends beyond the threat of a compromised SCADA system. To gain a more comprehensive picture, different types of cyber-attacks, including denial of service, malware, routing attacks, and protocol attacks must be considered [2]. These attacks may originate in the form of outsider attacks, insider attacks, operator errors and may be isolated or coordinated in nature [2]. One ramification of the diversity of cyber-attack methods is that the trustworthiness of the data the IDS receives can be violated through a variety of modes and locations within a wind power system. Corruption is not limited to the wind farm's SCADA system, but may be present in other areas (e.g. a turbine's supervisory controller). Researchers in [6] have described a similar situation where a nuclear turbine's controller has been infected with a Trojan horse designed to modify, replace, or nullify information forwarded to the intrusion detection systems.

The present situation of intrusion detection systems in wind power systems has motivated the current research. Presented in this paper is a distributed, model-based intrusion detection algorithm that has the ability to identify the presence of certain parameter manipulating cyber-attacks. The algorithm has been evaluated with simulation and its performance quantified. The focus of the paper will be the development, testing, and evaluation of the algorithm's performance on a test case using the Horns Rev wind farm in Denmark.

In Section 2 of this paper, a literature review of related work is presented. Section 3 details background information useful for understanding the rationale of the intrusion detection algorithm. Section 4 describes the intrusion detection algorithm in detail. Section 5 presents the Horns Rev wind farm and its use in the simulation development. Section 6 describes the functioning and development of the simulation. Results from simulation are presented and analyzed in Section 7. Section 8 offers a final conclusion on the project.

2. Literature Review

Intrusion detection systems for wired networks is a mature research area, whereas publications concerning IDS for wireless networks have only come about within the last decade. Authors in [7] introduced one of the first published techniques for applying IDS to wireless sensor networks. In their work, it was shown that a technique, denoted by the authors as spontaneous watchdogs, could serve as a general IDS framework to the application of static sensor networks. The paper discussed how, in certain situations, spontaneous watchdogs (dedicated sensor nodes) could serve to optimally watch over the communications of the sensors in a localized neighborhood.

In the past few years, many more IDS techniques have been proposed for both established and ad-hoc wireless networks. These techniques have included ideas such as locally or globally cooperative node networks and reputation/trust schemes. The author in [8] details a popular idea of using a distributed IDS where each node is a fully independent IDS. Unlike [7] where only certain nodes are responsible for network monitoring, this distributed scheme has nodes work in collaborative manner by exchanging information locally between neighbor nodes. Once an intrusion has been detected, the nodes in a localized area collectively decide on the response action.

Authors in [9] have introduced a cooperative, reputation-based scheme for MANETs (mobile ad-hoc networks) called Trust Evaluation and Reputation Exchange for Cooperative Intrusion Detection (TEREC). The aim of their research was to develop an IDS that isn’t reliant on a specific routing path for the nodes. To accomplish this, every node monitors its directly connected neighbors and reputation information is established and disseminated throughout the network based on the trustworthiness of the node’s measurements.

Similar to work in both [8] and [9], researchers in [10] have proposed a fully distributed anomaly detection system rooted in a trust and reputation scheme. The type of detection system detailed is a hybrid between anomaly and specification-based detection systems where specific parameters are configured within the system at initialization. This technique is partially unique in that it is a two-tiered system. Local nodes communicate and can deem other nodes suspicious through a Local Detection Engine, but a Global Detection Engine exists for a set of local nodes to appeal to if no consensus can be reached locally.

The critical nature of the SCADA systems has prompted significant research in their protection and resilience to cyber-attacks. Authors in [2], [3], [11] have detailed how compromise or destruction of SCADA systems for power systems can cause immediate physical and economic damage. Addressing wind power systems in specific, researchers in [3] have developed credible attack vulnerabilities within the SCADA system and detail some of their potential effects. Through simulation and modeling, the authors demonstrate how a malicious parameter manipulation of the power reference command can result in overspeed of a wind turbine.

As identified in [4], there are two types of attacks that modern SCADA systems are susceptible to: (1) packet access on networks encompassing SCADA devices (2) unauthorized access to control software or controllers. In the first attack scenario, the protection of the SCADA is derived from traditional cyber-security solutions such as firewalls, encryption schemes, and best practices. For instance, researchers in [8] have put forth wind energy specific security policy guidelines for the assembly and use of the IT components in the SCADA system. While these techniques provide a degree of security, authors in [4] identify situations, such as distributed, resource restricted networks, that these traditional solutions do not provide adequate security.

An excellent summary and classification of intrusion detection systems for SCADA systems is provided by authors in [12]. In this paper, the researchers categorize the existing IDS for SCADA systems into the various approaches: signature, anomaly, probabilistic, specification, and behavioral. They further go on to explain what the underlying mathematical basis is for each category as well as detailing whether each IDS is capable of detecting only know or unknown cyber-attacks.

Similar to the model-based solutions described in [12], authors in [13] describe an IDS for detecting attacks on SCADA Modbus protocols, OS platforms, and on networking infrastructure. Since SCADA networks tend to have fairly static, regular traffic patterns, they are excellent candidates for model-based IDS. The authors detail how they characterize the system for the expected behavior and identify attacks by detecting when the system deviates from the model.

Authors in [4] have proposed an IDS for distributed SCADA systems that incorporates the reputation/trust, fully distributed, and model-based techniques. In this paper, the researchers detail how nodes work collaboratively to build reputation information that is disseminated globally in order to isolate the misbehaving nodes within the network. While the authors employ the use of a model-based IDS framework, it is unique in that the system uses information gained from the application layer as opposed to from the routing or communication layers. This is achieved by creating a self-organizing map (SOM) that analyzes the spatial and temporal evolution of sensor readings.

Authors in [2] outline a similar, but more general approach to the application layer model-based design in [4]. The authors detail how a critical need for the future of cyber-security is the development of intrusion detection systems that incorporate the dynamics of the physical power system and the operational control structure. While only a framework, not a particular solution, is offered. They do provide reasoning as to why captivating domain-specific system behavior can create better security solution. Such a solution is, by definition, not generalizable across heterogeneous applications, but in exchange a security solution is created that forces an attacker to have an appreciable understanding of the system's dynamics in order to cause significant harm.

The present situation of intrusion detection systems in wind power systems has motivated the current research. Presented in this paper is a fully distributed, reputation- and model-based intrusion detection algorithm that has the ability to identify the presence of certain parameter manipulating cyber-attacks within a wind farm. More specifically, the solution provided is an IDS algorithm embracing the framework presented in [2] that takes advantage of two system properties unique to wind farms: 1) the spatiotemporal coherence of wind speed within a wind farm 2) the finite state machine structure of a wind turbine’s supervisory control logic. The proposed IDS is unique in that it is envisioned to be implemented in a distributed, wireless manner, but need not be resource limited like typical wireless sensor networks due to its application to wind turbines – a constant source of power. Similar to work in [4], but unlike most distributed IDS for wireless sensor networks, the proposed IDS leverages insight gained from the application layer (i.e. the wind farm properties mentioned above) as opposed to from the routing/communication layers.

3. Background

As mentioned above, the intrusion detection algorithm presented in the paper is a model-based design that takes advantage of two system properties unique to wind farms: 1) the spatiotemporal coherence of wind speed within a wind farm 2) the finite state machine structure of a wind turbine's supervisory control logic. In order to clarify why using these properties is advantageous, this section will present the necessary background on wind power systems. To ensure the reader has a sufficient understanding, this section will detail the control and operation of both individual wind turbines and entire wind farms. Additionally, the aerodynamic interaction amongst turbines, with specific focus on wind speed deficit, will be presented.

3.1 Wind Turbine Operation and Control

The basic premise behind a wind turbine is to capture the kinetic energy available in wind, convert it to rotational kinetic energy of the blades and hub, then convert this energy to usable electrical energy through a generator [14]. Throughout their history, wind turbines have taken on various designs and sizes, but the focus of this project will be on pitch controlled, horizontal-axis wind turbines (HAWT). Most, but not all, modern utility-scale wind turbines are of this configuration and numbers are expected to grow with new installations [14]. Below, in Figure 1, is an image of a representative HAWT with a few key components highlighted.

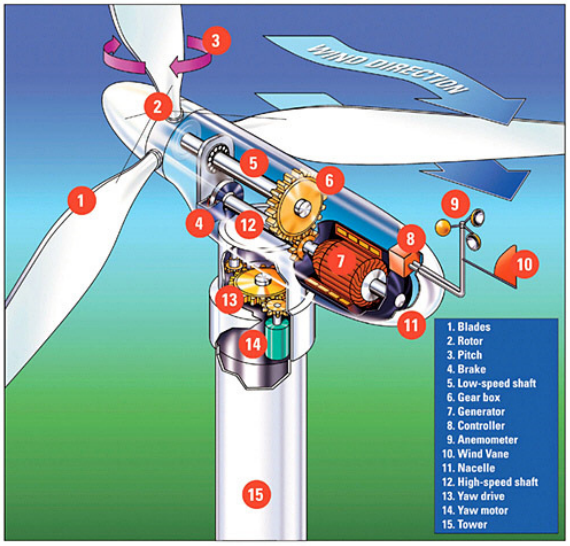


Figure 1: Typical horizontal-axis wind turbine [15]

The operation of a wind turbine is based on the time averaged wind speed the turbine measures through a meteorological device attached to the nacelle, which, most commonly, is an anemometer [5]. Wind speed is broken up into three regimes: below wind speed cut-in, allowable wind speeds, and above cut-out speed. When the averaged wind speed is below the cut-in speed or above cut-out speed, it is commanded to remain in an idle state. When the wind speed is in between these thresholds, then the turbine is free to spin and generate electricity

[5]. Typical cut-in velocity is about 3 or 4m/s and typical cut-out is about 20 to 25m/s, but exact values will depend on the specific turbine manufacturer [5]. There is also a rated wind speed, which is the particular wind speed that will allow the WT to produce its fully rated power [5].

Within a singular wind turbine, there are two layers of control: supervisory control and component control. These control systems consist of the various hardware, software, sensors, and actuators necessary to perform their function as well as some additional sensor information useful for monitoring but not necessary for operation [5]. The supervisory controller's job is to do top-level control where it is in charge of starting and stopping the turbine based on the wind regimes described above [5]. In Figure 2 below, a simplified state diagram is depicted that is representative of the core of most supervisory controllers.

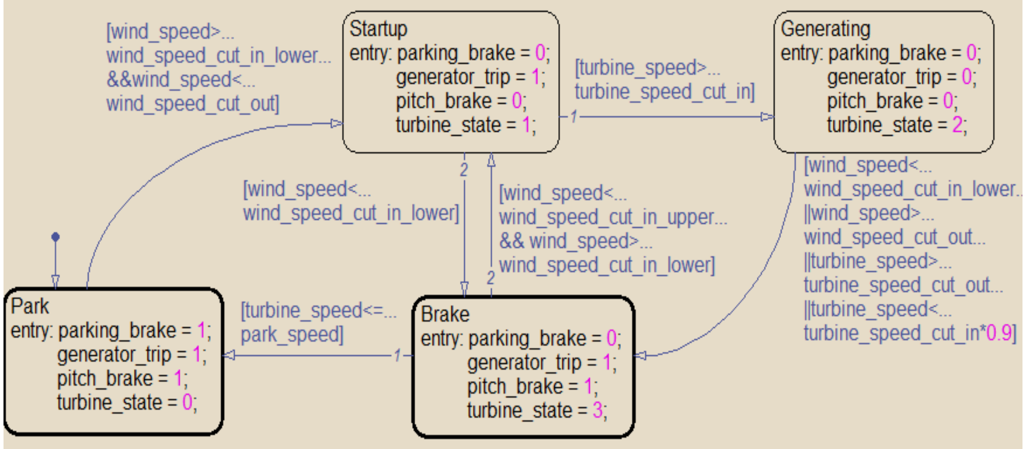


Figure 2: The Finite State Machine Structure of a Turbine Supervisory Controller [16]

Transitions between the states is dependent on the time averaged wind speed and wind speed cut-in and cut-out threshold parameters as well as turbine shaft speed and turbine shaft speed threshold parameters [17]. It can be seen that the supervisory controller is a finite state machine. There exist a finite set of operational states (e.g. park, startup, generating, and brake) the wind turbine can operate in where transition parameter criteria is determined by the manufacturer. Figure 3 below gives a more realistic portrayal of the states within a supervisory controller, but without the transition rules. In addition to dictating operating states, the supervisory controller is also responsible for giving set point values to component controllers, communication with wind farm operators through a SCADA system, and collection and reporting of sensor data [17].

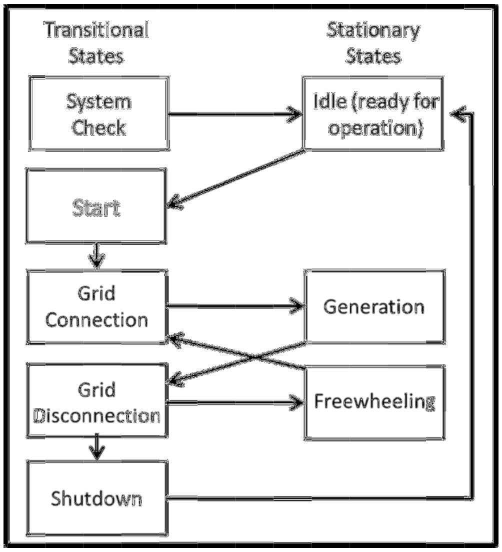


Figure 3: Transition Structure of Representative Supervisory Controller [17]

Component controllers, also called dynamic controllers, are the controllers that make continuous high-speed adjustments to specific subsystems such as the pitch, yaw, and generator systems [17]. For instance, the pitch controller is responsible for the pitch angle of the blades. It performs closed-loop control to keep the blades' pitch angles at the desired position and contains all of the necessary information for this control such as position and speed saturation rate parameters [17]. While each component controller is independent from other subsystem component controllers, their collective control is coordinated by the supervisory controller to achieve the desired overall control performance [17]. A simplified, yet informational, image of component control structure for a pitch controller is presented below.

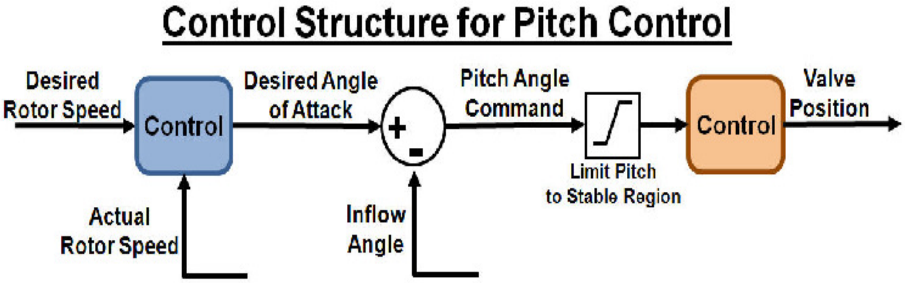


Figure 4: Representative Control Structure for Pitch Control [16]

So far, it has been detailed how any one particular wind turbine's control systems function through supervisory and component controllers, but it is also important to consider control of a wind farm as a whole. The image below shows the interaction and hierarchy of the three control levels for a wind turbine: component, supervisory, and farm level. Control of the wind farm is conducted by the farm operator through the SCADA system. The farm operator's responsibilities include: monitoring data reported by the turbines, updating the supervisory controllers with new software or operational set points, and shutting down/starting up turbines for various reasons such as maintenance [17].

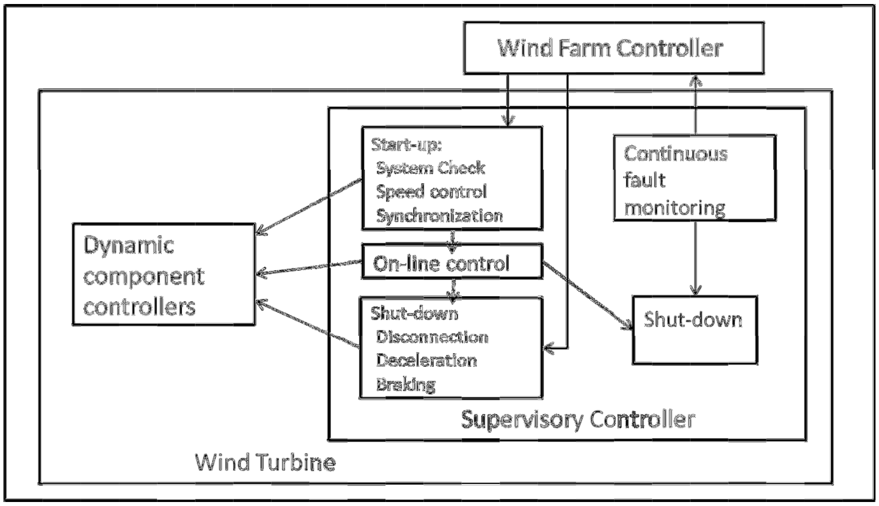


Figure 5: Control Structure Hierarchy [17]

Wind farm control could be considered to be nested inside an even larger control system of electrical grid power [18]. The control objective within this loop would be to continuously balance the instantaneous electrical demand and supply using all available power generating devices (e.g. wind power and conventional). Unfortunately, wind power is an energy source that is not conducive to central control due to the stochastic nature of wind fluctuations [18]. Researchers have addressed this issue by developing control schemes, such as delta or interleaved regulation, that control the output power from wind farms [18]. The basic premise of these control schemes is to have some or all of the wind turbines within a wind farm deliberately under-utilize the available wind energy so that there is always a power reserve. While these control schemes are abundant in literature, they are not as prevalent in application due to wind power's lack of penetration to the power grid. Instead of implementing a control scheme, transmission system operators (TSO) will often consider wind power as a fuel-saver. Meaning, the TSO will schedule conventional-energy plant production each day according to the demand while ignoring potential contribution from wind power. Any contribution from wind power is considered a negative demand load, thus acting as a fuel saver for traditional sources [18]. This is not the case when wind power begins to contribute a larger percentage of energy to the grid, but for this project it will be considered as such.

There are many more things to consider concerning the integration of wind power to the electrical grid and subsequent control over the farm, but these are beyond the scope the research in this project. The idea of control over a wind farm in context of the electric grid was brought up to rationalize an assumption made in this research: a wind farm will always seek to maximize its power output with the available wind. Meaning, no control schemes or commands from wind farm operators, such as power throttling to some or all wind turbines, will be considered.

Concerning the communication structure of wind farms, it was mentioned that the connection between the farm operator and the individual turbines is through a SCADA system.

A SCADA system is the collection of hardware and software that allows a centralized operator to bi-directionally communicate with the system components [5]. SCADA systems are not unique to wind power systems and have been previously used in many other applications such as industrial process control. Below is an image that shows some of the various configurations that a SCADA system could take for a wind power system.

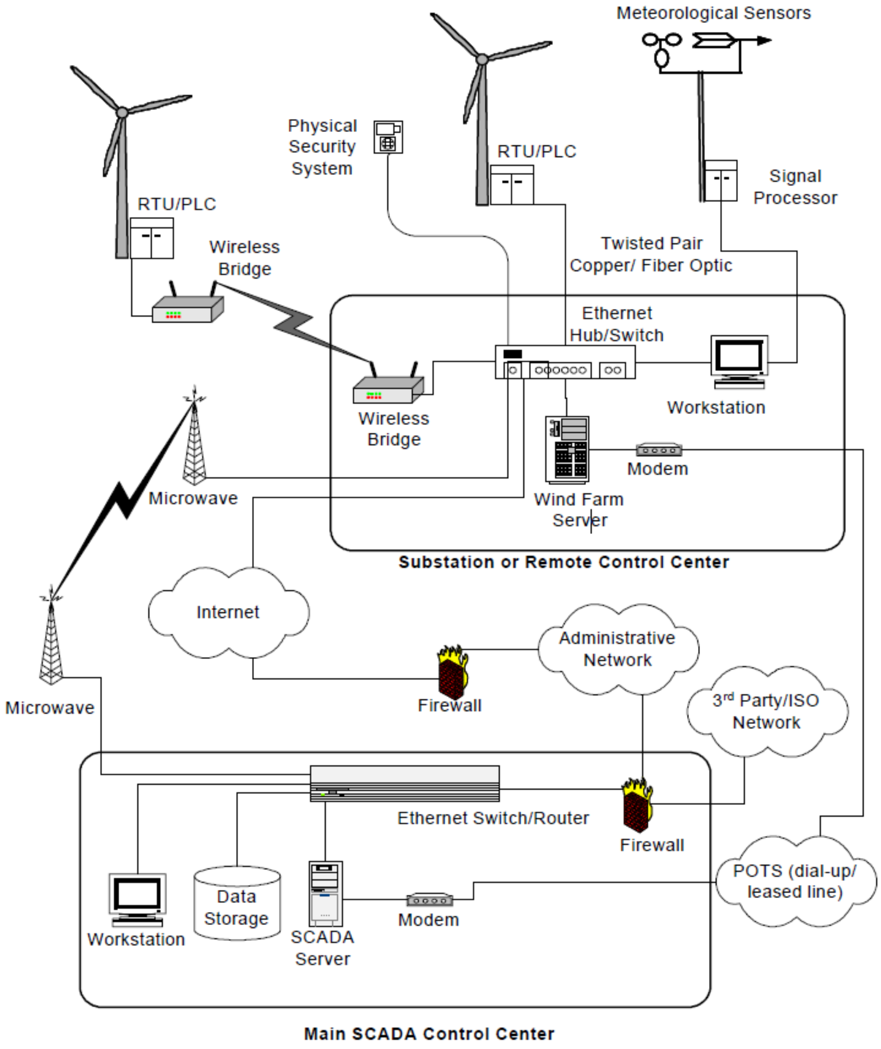


Figure 6: Potential SCADA configurations for a wind farm [11]

3.2 Aerodynamic Properties of Wind Farms

The aggregate nature of wind turbines within a wind farm gives rise to an interesting and complex situation concerning the movement of wind through a wind farm. Research on this topic has primarily focused on offshore wind farms. The reason for this is the difficulty involved in modeling the behavior of wind over irregular terrain surfaces and non-symmetric siting of wind turbines. Offshore wind farms are more conducive to analysis because they typically contain homogenous wind turbines sited symmetrically in a grid pattern within a topologically smooth area (i.e. a large body of water).

As mentioned above, the purpose of a wind turbine is to extract the kinetic energy from the wind and convert it to electrical energy. The effects of the kinetic energy extraction can be seen in a wind speed deficit that exists within the wake formed in the shadow of a turbine. In the figure below, the results from a computational fluid dynamics (CFD) simulation displays visually how a single wind turbine impacts the wind speed of its surroundings [19].

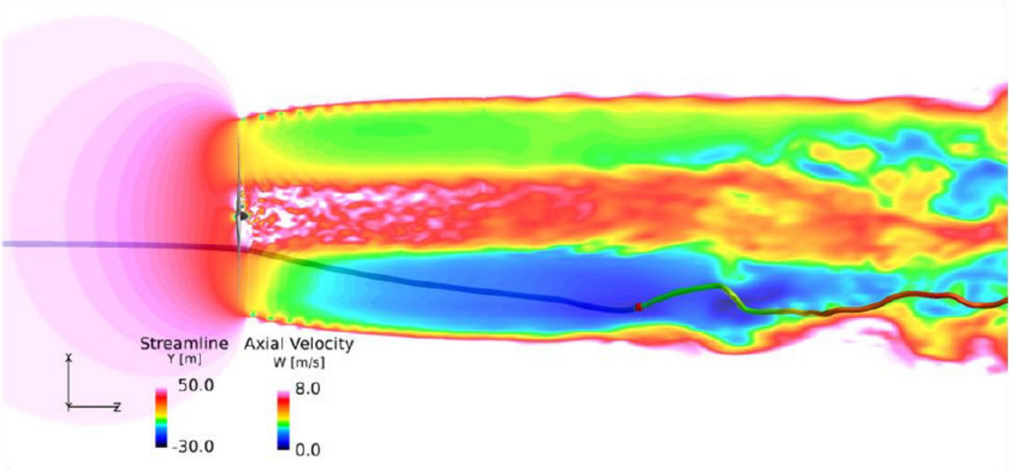


Figure 7: Visualization of wind speed deficit from a single wind turbine [19]

When a large number of turbines are grouped closely together, as they are in wind farms, the total power output from the wind farm is less than would be for a singular turbine multiplied by the number of turbines within the farm [17]. This drop in efficiency is due to turbines being eclipsed by the wakes of upstream turbines and is called array loss. Array losses in wind farms have been modeled through a variety of techniques, but are always a function of the turbine spacing within the farm, number of turbines, turbine controller operational parameters, wind turbulence intensity, and wind direction [17]. Figure 8 below is a photograph of the Horns Rev wind farm off the coast of Denmark that visually depicts the wakes of turbines in a wind farm.

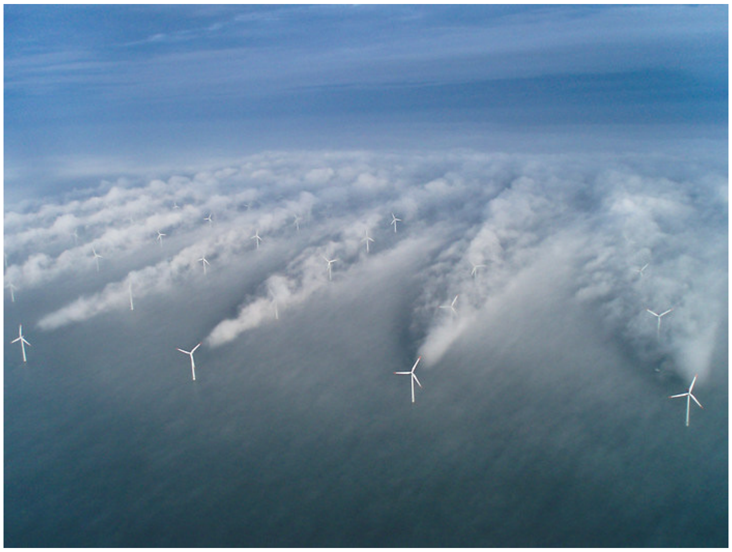


Figure 8: Visualization of wakes with Horns Rev wind farm. Photographer-Christian Steiness [19]

There are complex interactions between both the environment and adjacent wakes that help restore kinetic energy as the wakes propagate through the farm, but in general, greater wind deficits exists for turbines that are further downstream [17]. Array losses are between 5 and 20% of total potential output power of a wind farm [20]. Figure 9 below is a plot of the normalized wind deficits experienced along a row (array) of turbines within the Horns Rev wind farm. Figure 10 depicts the corresponding power loss for each array due to the wind speed deficits. The author in [21] constructed these graphs using SCADA data from the Horns Rev wind farm when the wind direction was 270±15˚. In context of the Horns Rev wind farm, 270˚ is wind directly from the west and would result in wake formation similar to Figure 8.

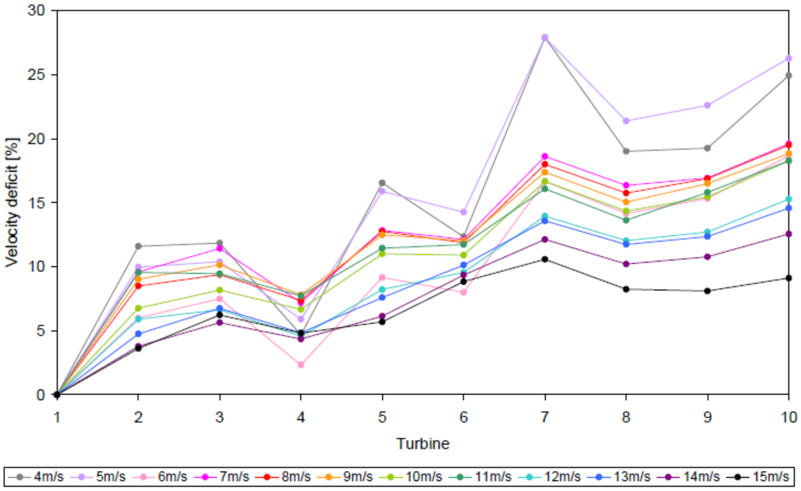


Figure 9: Velocity deficit along a row of turbines from west to east with winds from the west (270±15˚) [21]

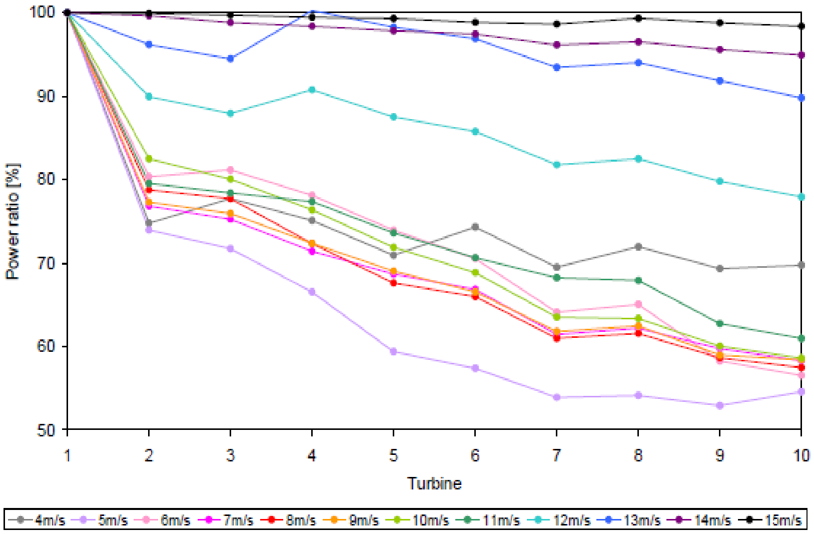


Figure 10: Array loss along a row of turbines from west to east corresponding to the wind deficits in Figure 8 [21]

Wind deficits and the finite state structure of wind turbine supervisory controllers create an interesting effect concerning the operation of wind turbines. In a scenario where the initial wind speed is zero, as wind moves toward the wind farm it first will reach the most upwind array of turbines. Once the wind speed has surpassed the wind speed cut-in threshold for the generation state of the turbine's supervisory controller, then that first array of turbines will begin to produce power [17]. If the wind speed is near the threshold value, then the wind deficit created by the operation of the first array of turbines will be sufficient to not allow subsequent downwind arrays to operate [17]. Only after the wind speed is high enough to compensate for all of the wind deficits from wind turbine wakes will the entire wind farm be in the generation state. Figure 11 illustrates the coupling between wind speed and the state of the turbines. On the left of the figure is a plot of the wind speeds as reported by turbines in the Horns Rev wind farm. On the right is a corresponding plot of the power production at those wind speeds. It can be seen that the wind speed was near the wind speed cut-in threshold of 4m/s for the turbines in Horns Rev. This illustrates why downwind turbines tend to produce less (or no) power than those that are upwind.

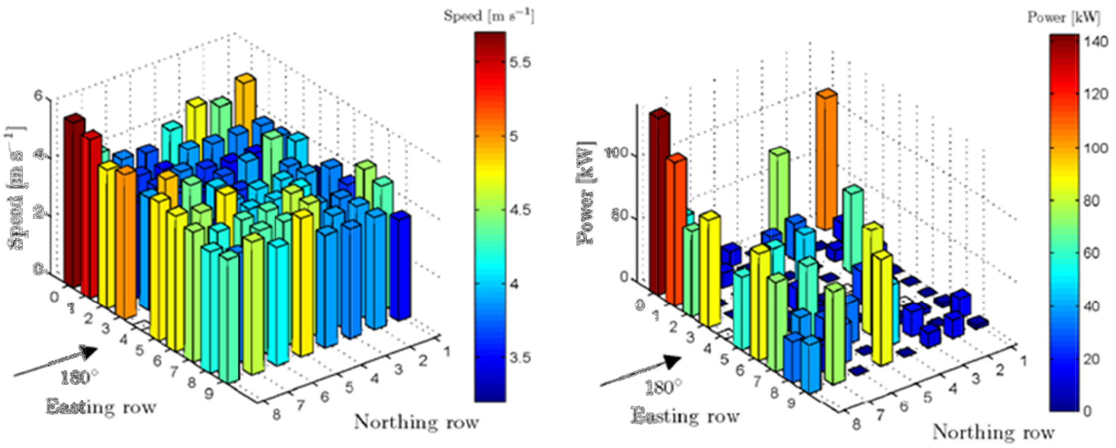


Figure 11: A pair of plots displaying the effect of wind deficits on the operational state of turbines in the Horns Rev wind farm [19]

4. Intrusion Detection Algorithm

This section will present the details of the distributed, reputation and model-based intrusion detection algorithm that has the ability to identify the presence of certain parameter manipulating cyber-attacks within a wind farm. It is a model-based algorithm specifically designed for wind farms, because it takes advantage of the spatiotemporal coherence of wind speed within a wind farm and the finite state machine structure of a wind turbine's supervisory controller. Rationale for the implementation of a reputation scheme has to do with the stochastic nature of the dynamic system governing the states of the turbines (i.e. wind). In the background section above, it was discussed how wind speed within a wind farm is spatially and temporally correlated, but that the models are not deterministic due to wind's turbulent behavior. A reputation scheme is advantageous for this situation because it permits temporary disagreements in state amongst turbines. This property is useful in that it helps alleviate the issue of false detection from turbine state disagreements due to the natural, stochastic fluctuation in wind.

As a simplifying assumption, the IDS presented here will limit the states of a turbine's supervisory controller to the generation and idle states. Referring back to Figure 3, it can be seen that these are the two primary stationary states as the freewheeling state is often considered part of the idle state. As mentioned in [17], the time spent in transitory states is insignificant relative to stationary states, so this is not an unreasonable assumption. In addition, it will be assumed that the turbines within the wind farm are spaced in a regular, grid-like manner as they are in virtually all large offshore wind farms (e.g. Horns Rev and Nysted).

The algorithm's goal is to identify the presence of anomalous wind turbine behavior within a wind farm and is reliant on a reputation scoring scheme to do so. In this IDS algorithm, each turbine has a reputation score bounded between 0 and 100 that is effectively an indicator of how often it agrees in state (i.e. the state of the turbine's supervisory controller) with the states of turbines nearest in physical proximity. Turbines initially are assigned a perfect reputation score of 100, but have their scores manipulated at regular intervals. The length of this interval is set to the duration of the turbine's supervisory controller wind speed sample period which is manufacturer dependent. Typically, this length is between 1 and 5 minutes [5]. A threshold reputation score is defined such that, when a particular turbine's reputation score drops below said threshold, it will be deemed anomalous. By anomalous, what is meant is that a particular turbine has operated in such a manner that it has become suspicious that it is operating properly in context to the operation of the turbines located physically nearby. In this way, the IDS detailed here is of the (locally) collaborative type that relies on the "wisdom of the crowd" effect: if a turbine is consistently in a different state from its neighboring turbines, then there is reason to suspect malfunctioning.

If we consider a wind farm where the turbines are oriented in a symmetric grid as they are in Figure 12, local collaboration is defined as interaction between a turbine of interest and the eight adjacent turbines. Turbines that reside on an edge of the farm will interact with the adjacent turbines that do exist. For convenience, the eight turbines surrounding a central turbine will be defined as that turbine's neighborhood. Figure 12 visually depicts the neighborhoods of node 5 and 10.

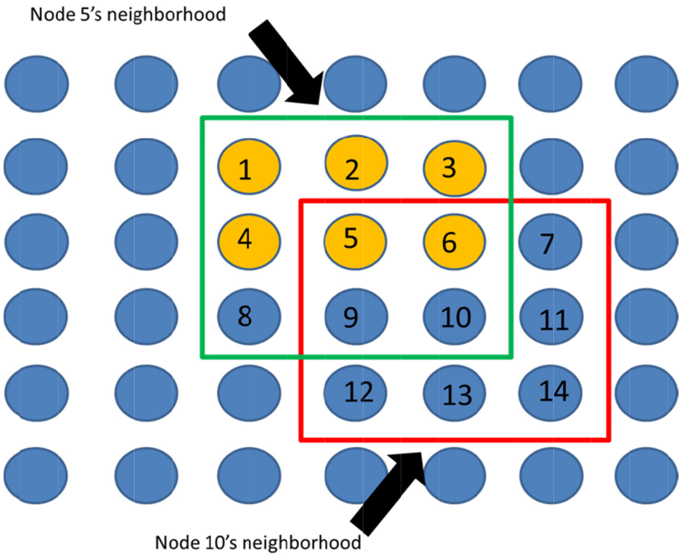


Figure 12: Image that illustrates the definition of a neighborhood within a wind farm

The adjustment of a turbine's reputation score is the result of the net increment/decrement commands sent by neighboring turbines. A bonus will be added to a turbine's reputation score for every neighbor that it agrees in state with, while a penalty will be subtracted from its reputation score for every neighbor turbine that it does not share a state with. This score aggregation process takes place at regular time intervals according to the length of the supervisory controller's wind speed averaging length mentioned above. A visual aid of the collaboration process is provided in Figure 13. Assume the blue nodes represent the generation state of a wind turbine's supervisory controller, while orange nodes represent the idle state. For this example, let the bonus equal +1 and the penalty equal -1, and note that the node of interest is in the generation state. Each of the node of interest's five neighbors that share the generation state with it will direct the turbine of interest to increase its reputation score by +1, while each of the idle state turbines will direct it to decrease by -1. The net result is a +2 increase (5-3=2) to the node of interest's reputation score.

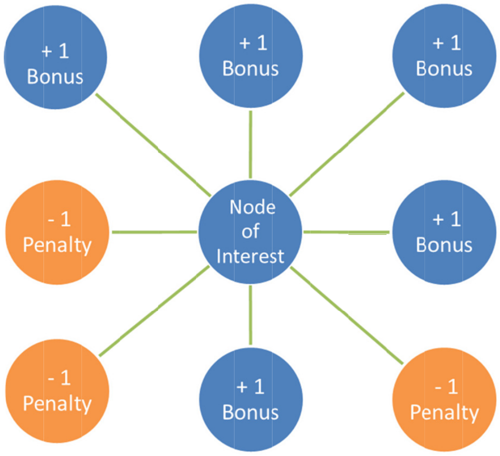


Figure 13: Depiction of local collaboration and reputation adjustment

A mathematical description of the reputation scoring scheme for a given turbine is shown below. A turbine's reputation score at time t is its reputation score at time t-1 plus the contributions from its neighbors. The number of neighbors a turbine has is dependent on its location within the wind farm. A neighboring turbine's contribution is the product of the nominal penalty or bonus value times its scaled reputation score at t-1. By weighting the contribution of a neighbor turbine according to its reputation score, the efficacy of untrustworthy turbines (those with low reputation scores) is reduced. The nominal bonus and penalty values can be unequal and are up to the user to set. For a more "aggressive" algorithm that can detect corruption quicker, but with the tradeoff of a higher false detection rate, the nominal penalty value can be set to a higher value than the nominal bonus value. The absolute and relative values of the nominal bonus and penalty values affect the behavior of the algorithm and this behavior is discussed more in the results section below.

$$\text{reputation\_score}_t = \text{reputation\_score}_{t-1} + \sum_{i=1}^{\#\,of\,neighbors} \text{ith\_neighbor\_contribution}$$

Contributions from a neighboring turbine that shares or does not share the same state, respectively:

$$\text{ith\_neighbor\_contribution} = \text{nominal\_bonus} * (\text{neighbor\_reputation\_score}_{\mathrm{t-1}})/100$$

$$\text{ith\_neighbor\_contribution} = \text{nominal\_penalty} * (\text{neighbor\_reputation\_score}_{\mathrm{t-1}})/100$$

5. Horns Rev Wind Farm: A Case Study

The Horns Rev wind farm is a large offshore wind farm located on a natural reef in the North Sea 15 km west of Denmark [22]. It consists of 80 Vestas 2MW wind turbines and was built by the Danish utility group Elsam in 2002 [22]. The Horns Rev wind farm was selected as the test bed for the simulation of the intrusion detection algorithm presented in this paper. The Horns Rev wind farm is the most prominent offshore wind farm in literature, and researchers have investigated nearly every aspect of its performance. It is this abundance and availability of wind farm system properties that Horns Rev was chosen for a case study.

There are a few specific properties of Horns Rev that are pertinent to development of the simulation that is presented in the next section. These properties are:

- Physical orientation of wind turbines within the farm (Figure 14):
    - 8 rows by 10 columns of turbines
    - 560 meters spacing between turbines
    - Farm forms a rectangle with axes oriented north-south and east-west
        - This is a close approximation – it reality, it is a slightly skewed rhombus
- Vestas V-80 2MW wind turbine supervisory controller:
    - 4 m/s cut-in wind speed threshold for generation state
    - 25 m/s cut-out wind speed threshold for generation state
    - 5 minute duration for wind speed averaging
- Wind farm wind properties:
    - Wind deficits from wake losses as a function of wind direction (Figure 9)
    - Description of free stream wind speed characteristics (Figure 15)

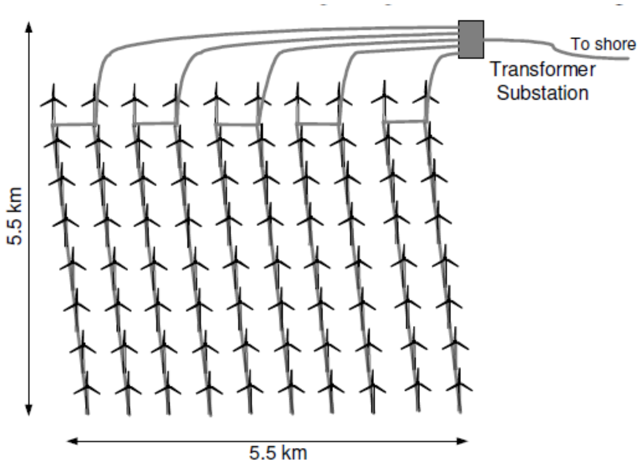


Figure 14: Physical orientation of Horns Rev wind farm [22]

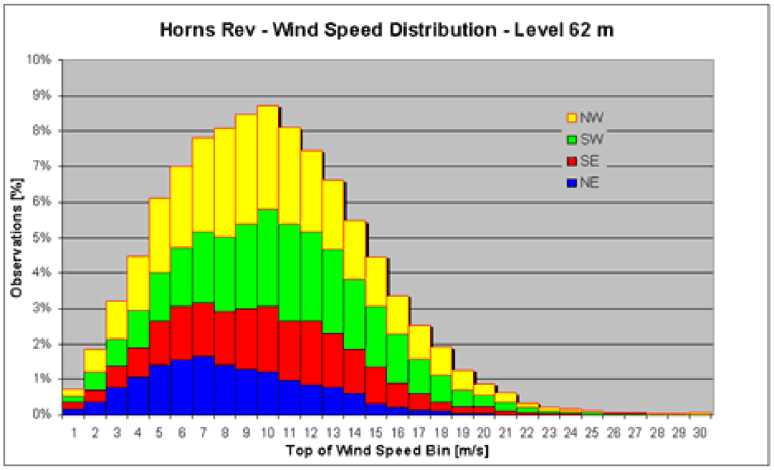


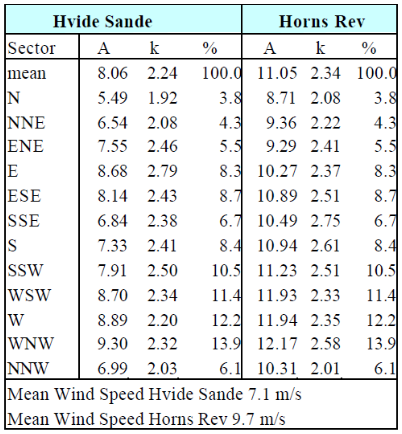
Figure 15: Weibull distribution of free stream wind speeds at Horns Rev wind farm [23]

6. Simulation Development

This section will discuss the development and functioning of the Matlab simulation designed to test the intrusion detection algorithm. Recalling from above, the intrusion detection algorithm relies on state information from turbines. In a real world scenario, the wind turbines would transition between states through time according to the particular wind environment they experience. Since the intrusion detection algorithm must be tested in simulation, it is necessary to develop a synthetic series of turbine states. The main function of this simulation is to develop a spatiotemporal profile of wind turbine states and retroactively apply the intrusion detection algorithm to search for corrupted turbines. It should be noted that the retroactive application of the intrusion detection algorithm is an artifact of simulation and the simulation should still approximate how the algorithm would perform in a real wind farm. The development of a spatiotemporal profile of wind turbine states requires the following set of information: a time series of free stream wind speed, wind turbine supervisory controller parameters, wind turbine spacing and orientation within a wind farm, dynamic behavior of wind within the wind farm, and corrupted turbine behavior and location. The subsections below will detail how each piece of information is used and integrated to create the simulation.

6.1 Wind Speed Time Series

A time series of free stream wind speed is the data that describes the wind speed prior to interaction with the wind farm. The simulation detailed in this project requires a time series of wind speed as an input. To help develop a realistic a simulation, real values describing the wind speed characteristics at Horns Rev were used. Wind speed is characterized according to the Weibull distribution in Figure 15, and specific directional parameters are detailed below in Figure 16.



Hvide Sande				Horns Rev		
Sector	A	k	%	A	k	%
mean	8.06	2.24	100.0	11.05	2.34	100.0
N	5.49	1.92	3.8	8.71	2.08	3.8
NNE	6.54	2.08	4.3	9.36	2.22	4.3
ENE	7.55	2.46	5.5	9.29	2.41	5.5
E	8.68	2.79	8.3	10.27	2.37	8.3
ESE	8.14	2.43	8.7	10.89	2.51	8.7
SSE	6.84	2.38	6.7	10.49	2.75	6.7
S	7.33	2.41	8.4	10.94	2.61	8.4
SSW	7.91	2.50	10.5	11.23	2.51	10.5
WSW	8.70	2.34	11.4	11.93	2.33	11.4
W	8.89	2.20	12.2	11.94	2.35	12.2
WNW	9.30	2.32	13.9	12.17	2.58	13.9
NNW	6.99	2.03	6.1	10.31	2.01	6.1
Mean Wind Speed Hvide Sande 7.1 m/s						
Mean Wind Speed Horns Rev 9.7 m/s						

Figure 16: Horns Rev's Weibull wind speed parameters by direction [23]

The generation of a wind speed time series cannot be attained by sampling the Weibull distribution, but must take into account the autocorrelation nature of wind. The program HOMER, developed by the National Renewable Energy Laboratory (NREL), is a software tool that can take in Weibull distribution parameters as inputs and output an appropriate, autocorrelated time series of wind speed. As an assumption for this project, wind is considered one dimensional. Meaning, wind will be considered to move directly from west to east. The time series produced by HOMER were exported to Matlab where they were used to assist in the generation of the spatiotemporal profile of turbine states. The image below is a time series of wind speeds produced by HOMER.

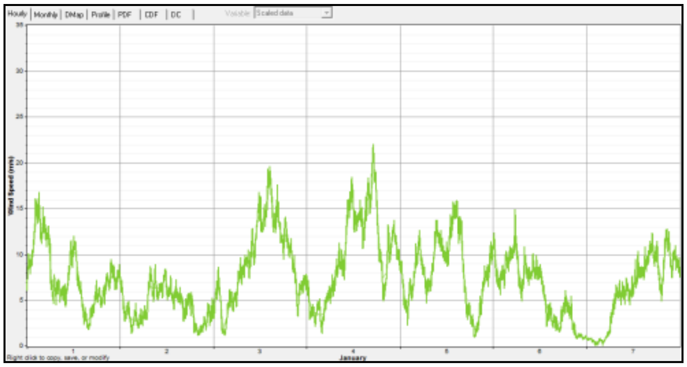


Figure 17: Time series of wind speed produced by HOMER software

6.2 Wind Farm Spatial Properties

The simulation requires the spatial orientation of all of the turbines within the wind farm. As discussed in Section 5, this simulation codes turbines in an 8 row X 10 column rectangular grid with 560 meters of spacing between turbines to mimic the Horns Rev wind farm. The spatial information is an important aspect of the simulation as it is necessary for determining turbine neighbors for the intrusion detection algorithm. Figure 18 is a visualization of how the turbines are spaced within the simulation.

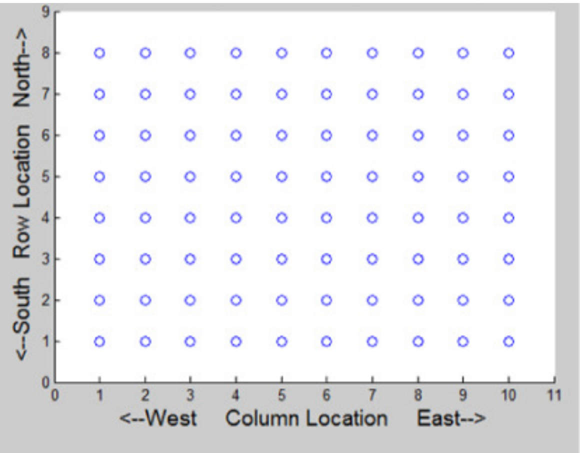


Figure 18: Visualization of turbine spacing within the simulation

6.3 Generating a Wind Speed Time Series for Every Turbine

To achieve the goal of generating a spatial and temporal profile of turbine states, it is necessary to have a time series of wind speed for every turbine. To do this, the free stream wind speed time series discussed in Section 6.1 must be modified for each turbine according to its position within the wind farm. There are two aspects of the modification. First, the time series must be temporally shifted to compensate for the time required for wind to propagate through the wind farm. For instance, turbines that are further downwind will experience a wind front a few minutes later than the upwind turbines. At 4 m/s, it takes about 15 minutes for wind flow to pass through Horns Rev [19]. Assuming the speed of propagation scales linearly, Horns Rev's average westerly wind speed of 9.6 m/s will take approximately 1 minute to travel between two columns of turbines. As an assumption, wind will always propagate at this rate. This information is used to temporally shift the wind speed time series for each wind turbine according to its location within the wind farm.

The second modification to the wind speed time series is the magnitude adjustment necessary to compensate for the wind deficits caused by turbine wakes. As discussed in Section 6.1, downstream turbines will experience will experience a lower mean wind speed than their upstream counterparts. Together with the temporal shift discussed before, these are the modifications considered for generating a time series of wind speeds for every turbine within the wind farm. The wind speed deficits used can be found in Figure 9 in Section 3.2.

6.4 Supervisory Controller Parameters and Corruption Emulation

The simulation is built such that a set of supervisory control parameters is stored for every turbine. Referring back to Section 3.1, a wind turbine's supervisory controller is a finite state machine that transitions between states under certain criteria. For this simulation, the state transition parameters of wind speed cut-in threshold, wind speed cut-out threshold, and the wind speed averaging duration are stored for each turbine. The values of these parameters are discussed in Section 5.

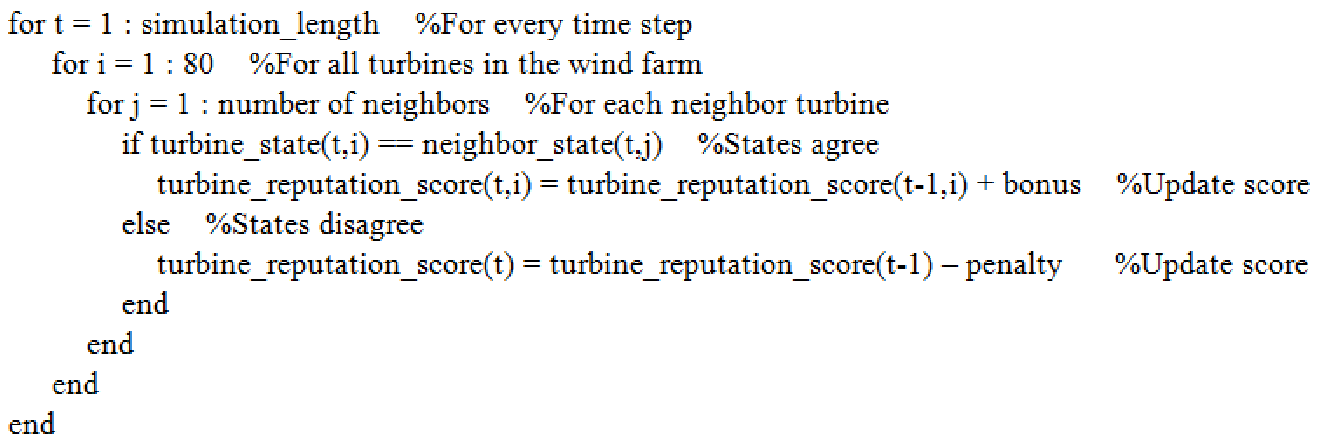
The type of cyber-attack considered for this project is a parameter manipulating attack to a wind turbine's supervisory controller. Motivation for this cyber-attack scenario was discussed in Section 1, and its importance highlighted by authors in [3] who have demonstrated the damaging effects of malicious control parameter manipulations in wind turbines. The mechanism for corruption emulation in the simulation is a change in value of the cut-in threshold, cut-out threshold, or averaging duration. Since each turbine has its own controller parameters, corruption can be mimicked in any number or location of turbines. The simulation requires as an input the locations and severity of parameter manipulating corruptions.

6.5 Generation of a Spatiotemporal Profile of Wind Turbine States

As discussed above, a spatiotemporal profile of wind turbine states is the information required to apply the intrusion detection algorithm. This profile is created by combining the wind speed time series adjusted for propagation duration and wind deficit with the supervisory control information stored in each turbine. The state of a turbine's supervisory controller is dependent on the wind conditions it experiences along with its state transition criteria. The combination of this information is what yields a spatiotemporal profile of wind turbine states. Since corruption is the modification of the turbine's state transition criteria, a corrupted turbine will yield a time series of turbine states that is different than would be if it were an unaffected turbine.

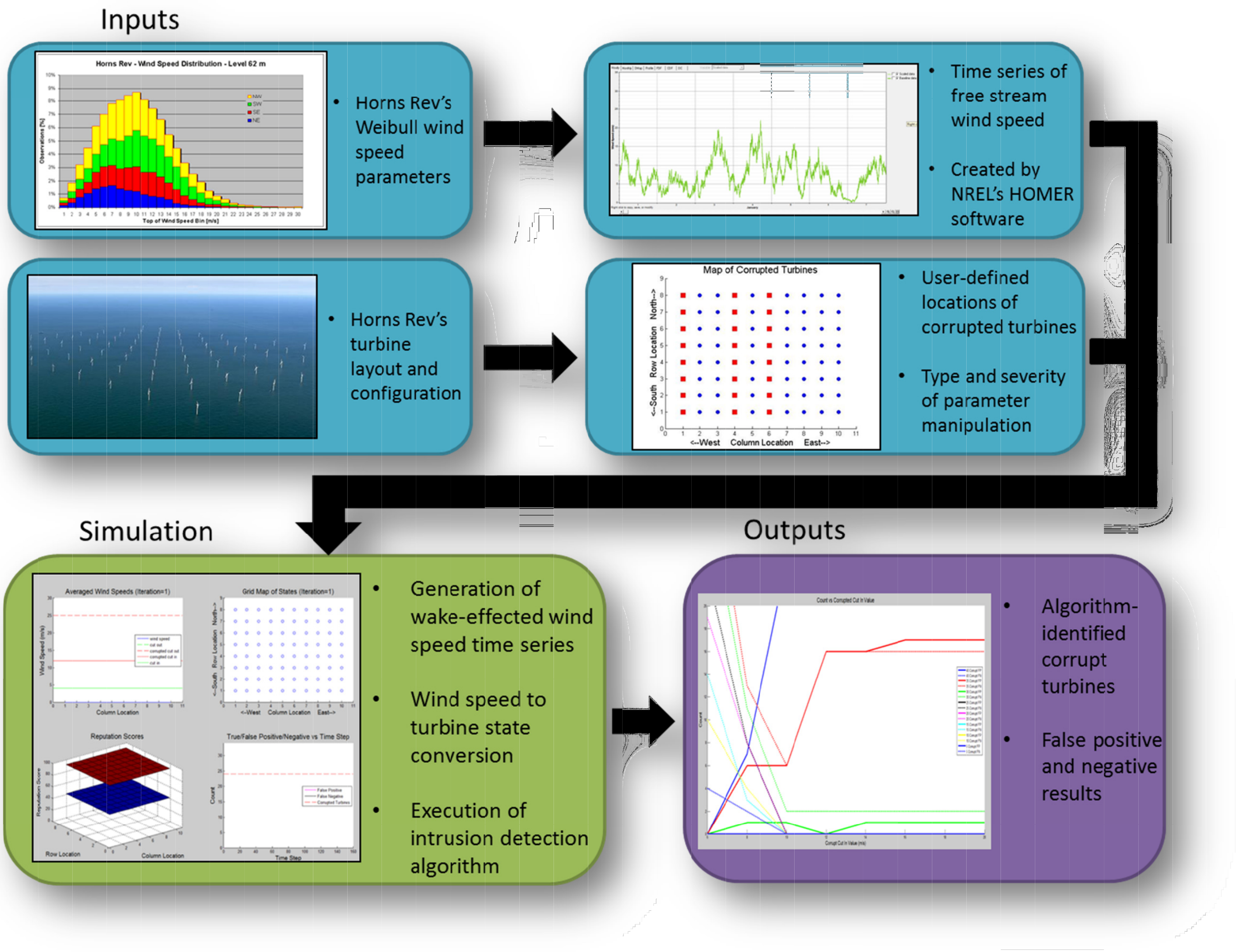
6.6 Application of the Intrusion Detection Algorithm

With the spatiotemporal profile of wind turbine states at hand, the intrusion detection algorithm can be applied. Repeating from earlier, the algorithm is applied retroactively as a matter of simulation convenience, but should not yield different results from what would be if implemented real-time. The algorithm was described in detail in Section 4, but its implementation in simulation is detailed here. The spatiotemporal profile of turbine states can be considered a three dimensional matrix with each two dimensional slice representing all of the states of the turbines at a given time. Essentially, it is a history of the evolution of turbine states through time. The intrusion detection algorithm is initially applied at the first time step where it performs reputation score manipulations according to the algorithm's details. Once the algorithm completes a particular time slice, it stores the reputation information and proceeds to the next time slice. This process is repeated iteratively until the entire profile is complete or it reaches a user-defined stop criterion (e.g. stop the simulation when X number of turbines have been identified). Matlab pseudocode for the algorithm is presented below. Figure 20 is an overview of how the simulation is organized.



```
for t = 1 : simulation_length    %For every time step
    for i = 1 : 80    %For all turbines in the wind farm
        for j = 1 : number of neighbors    %For each neighbor turbine
            if turbine_state(t,i) == neighbor_state(t,j)    %States agree
                turbine_reputation_score(t,i) = turbine_reputation_score(t-1,i) + bonus    %Update score
            else    %States disagree
                turbine_reputation_score(t) = turbine_reputation_score(t-1) – penalty      %Update score
            end
        end
    end
end
```

Figure 19: Matlab psuedocode for simulation



Inputs
Horns Rev - Wind Speed Distribution - Level 62 m
Observations [%]
Top of Wind Speed Bin [m/s]
NW
SW
SE
NE
Map of Corrupted Turbines
<--South Row Location North-->
<--West Column Location East-->
Simulation
Averaged Wind Speeds (Iteration=1)
Grid Map of States (Iteration=1)
Reputation Scores
True/False Positive/Negative vs Time Step
Outputs

- Horns Rev's Weibull wind speed parameters
- Time series of free stream wind speed
- Created by NREL's HOMER software
- Horns Rev's turbine layout and configuration
- User-defined locations of corrupted turbines
- Type and severity of parameter manipulation
- Generation of wake-effected wind speed time series
- Wind speed to turbine state conversion
- Execution of intrusion detection algorithm
- Algorithm-identified corrupt turbines
- False positive and negative results

Figure 20: Simulation overview

7. Results

The proposed intrusion detection algorithm has been simulated to evaluate its performance. A variety of attack scenarios were tested to display the algorithm's behavior in various corruption contexts. As discussed in Section 6.4, the type of corruption emulated was a parameter manipulation of a wind turbine's supervisory controller. Specifically, the generation state's cut-in wind speed threshold was altered from its nominal value of 4 m/s. The magnitude of the parameter shift, location of corrupted turbines within the wind farm, and the quantity of corrupted turbines were the test variables manipulated to display the algorithm's performance in a variety of attack scenarios. The metrics presented to evaluate the algorithm's performance are false positives and false negatives. For the simulation results presented below, the algorithm's nominal bonus and penalty values were 4 and 8, respectively. The absolute value and relative difference between these two values affect the algorithm's performance, but was held constant for consistency. How these values affect the algorithm's performance is detailed at the end of this section.

7.1 Baseline – No corruption

To begin, a simulation was run with no corrupted turbines to demonstrate the algorithm's baseline performance. Unless specified otherwise, this and subsequent simulation results all contained the system properties discussed in Section 5. The simulation was run for 160 iterations (800 minutes) where each iteration is 5 minutes. The image on the left of Figure 21 is a history of the false negatives and false positives generated by the algorithm. The image on the right of Figure 21 depicts the locations of corrupted turbines. Blue circles indicate non-corrupted turbines, and in later examples, red squares indicate the locations of corrupted turbines. From the baseline simulation, it can be seen that when there are no corruptions to the wind farm the algorithm produces no false positives or false negatives.

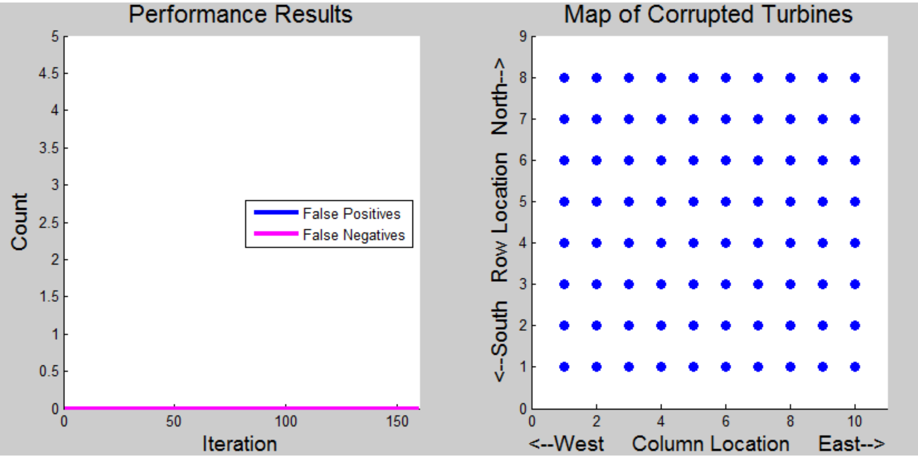


Figure 21: Baseline simulation results

7.2 A small set of corrupted turbines

In this simulation, the generation state's wind speed cut-in parameter was altered from 4 m/s to 12m/s for 5 turbines. The locations of these turbines were randomly assigned and are indicated on the right of Figure 22. From the performance results, it can be seen that the first few time steps produce 5 false negatives. This is normal behavior as it takes multiple iterations for individual turbines to have their reputation score decremented past the corruption threshold score of 50. The additional false negative around the 60th iteration is a result of turbines having the ability to regain trustworthiness. In other words, turbines are not permanently deemed corrupt and are considered OK if their reputation score increases pass the threshold. This property may not be desirable in a real-world scenario, but was deliberately kept in for transparency and to demonstrate the difficultly of deciphering between corruption and the stochastic nature of wind.

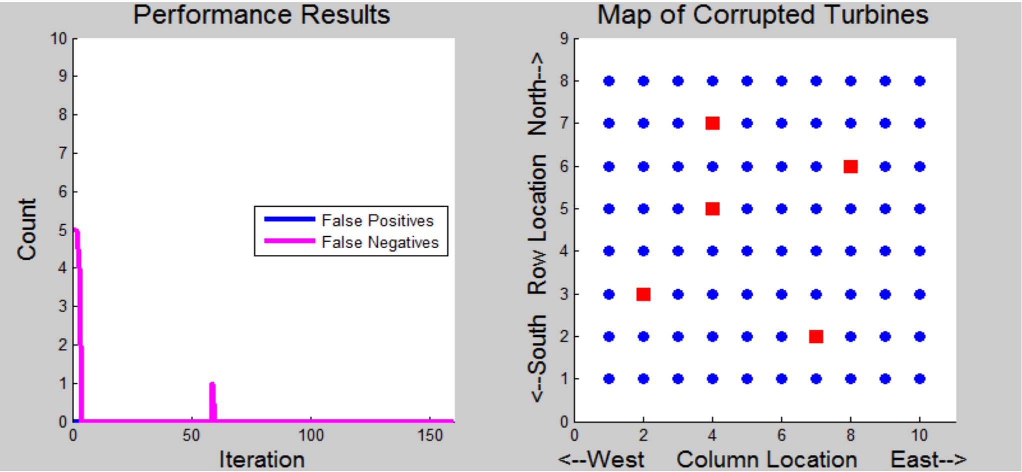


Figure 22: Results from a small set of corrupted turbines

7.3 Varying the level of turbine corruption

This test scenario demonstrates the algorithm's performance in context to the magnitude of the parameter manipulation. A fixed topology of 3 columns of corrupted turbines can be seen below in Figure 23. A total of 8 simulations were run with this corruption topography where the corrupt turbines' generation wind speed cut-in parameter was varied from 6m/s to 20m/s in 2m/s increments. The count of false positives and false negatives after 160 iterations is plotted against the corruption level. The algorithm produced a substantial amount of false positives at the 6m/s corruption level and a few at the 8m/s level, but produced none at higher levels of corruption. This is the expected trend because smaller, more subtle parameter manipulations are more difficult to detect.

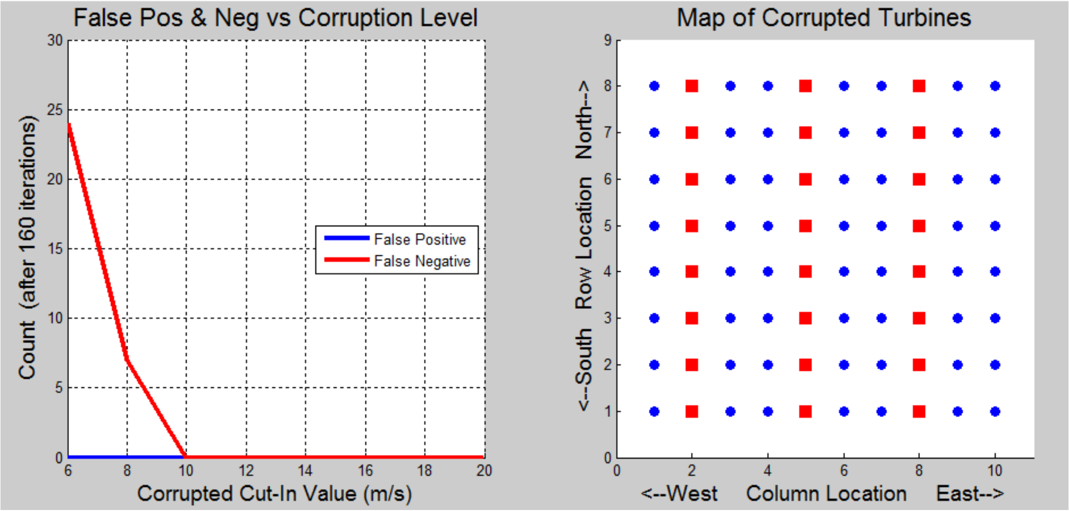


Figure 23: Results demonstrating the effects of varying the level of corruption

7.4 Varying the quantity of corrupted turbines

In this simulation, the level of corruption was held constant while the number of corrupted turbines was varied. The locations of the corrupted turbines were randomly assigned and the quantity of corrupted turbines varied from 5 to 40 with increments of 5 turbines. The manipulation of the generation wind speed cut-in parameter was held constant at 12 m/s. This simulation helps demonstrate the algorithms performance against different quantities of corrupted turbines. Figure 24 shows the history of false positives and false negatives for each quantity of corrupted turbines. It can be seen that when 30 or more turbines are corrupted there are false positives and negatives reported at every iteration. The performance of the algorithm degrades as the number of corrupted turbines increases. The algorithm is founded on the idea of “wisdom of the crowd,” so it is expected to perform poorly as the number of corrupted turbines approaches 40 (half of the total number of turbines at Horns Rev wind farm).

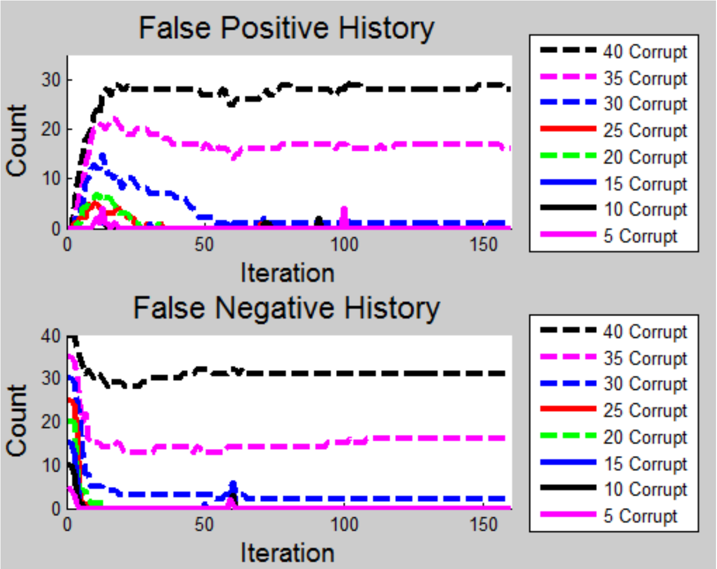


Figure 24: History of false positives and false negatives for different amounts of corrupted turbines

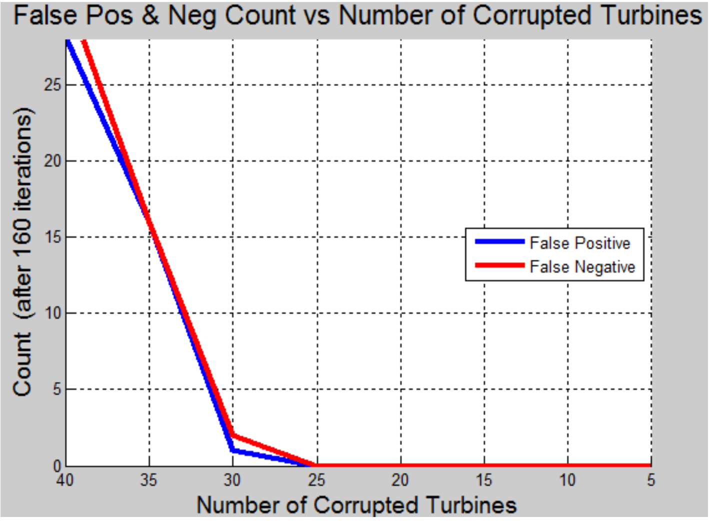


Figure 25: False positive and false negative counts at the end of 160 iterations

7.5 Varying the quantity of corrupted turbines and the corruption level

This simulation scenario provides a more complete demonstration of the algorithm's performance as the number and level of turbine corruptions is varied. Figure 26 depicts the false positive and false negative counts after 160 iterations for 64 different simulation settings. Similar to 7.4, the locations of the corrupted turbines were randomly generated and varied between 5 and 40 corrupted turbines. At each of these quantities, the level of corruption was varied from 6m/s to 20m/s. Congruous with results from before, the algorithm performs better when the quantity of corrupted turbines is low and the level of corruption is high.

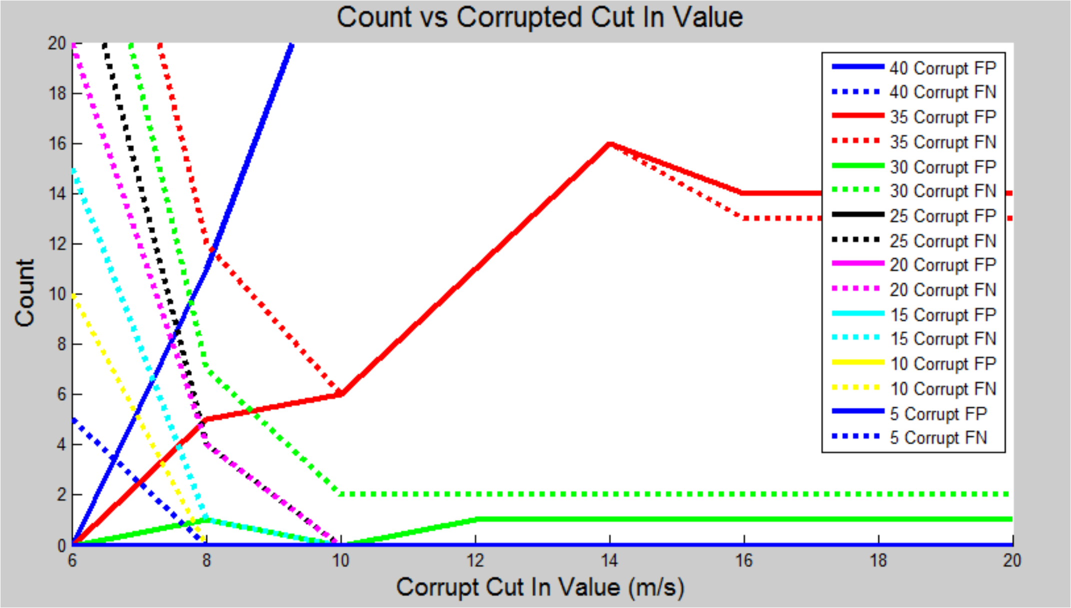


Figure 26: Results

7.6 Comments

It was found that the algorithm performed best when there were less than 30 corrupted turbines and the turbine's wind speed cut-in parameter was altered to a value of 10m/s or higher. When these conditions were present, the algorithm properly identified all corrupted turbines and yielded zero false positives or false negatives at the end of an 800 minute simulation. As the number of corrupted turbines approaches 40 (half of the total number of turbines at Horns Rev), the algorithm's performance degrades and begins to yield increasingly more false positives and false negatives. The algorithm is not applicable to situations when half or more of the turbines are corrupted due to its reliance on the "wisdom of the crowd" principle. When the magnitude of the wind speed cut-in parameter manipulation is small (value changed to 10m/s or less), the algorithm's performance degrades as the corrupted value approaches the nominal value of 4m/s. As the corrupted parameter value approaches the nominal parameter value, the corrupted turbine's state profile begins to more closely approximate the state profile that would be generated if that turbine were not corrupted. This sheds insight into why the algorithm has difficulty identifying turbines with subtle corruption.

As mentioned above, the nominal bonus and penalty values used for the algorithm were 4 and 8, respectively. These values were chosen for the simulation tests because they yielded reasonably good results, but should not be considered optimal values. Manipulations to these values would affect detection time, detection probability, false positive rate, and false negative rate. For instance, a small bonus and large penalty would create a more "aggressive" algorithm. It would be aggressive in the sense that it could more quickly identify a corrupted turbine, but with the tradeoff of a higher false positive rate. The exact values of the nominal bonus and penalty would be user-specified according to the user's desired algorithm characteristics.

8. Conclusion

There are many obstacles wind power must overcome to continue to grow as an industry, and one of the most important is protection against cyber-attacks. Researchers in [3] have demonstrated how malicious control parameter manipulations in wind turbine supervisory controllers can result in serious physical and economic damage. Similar to other cyber-physical power infrastructure systems, vulnerabilities to such a cyber-attack are found in the form of insider, supply chain, man-in-the-middle, and various other attack methods. Few publications exist that provide cyber-security solutions for wind farms in specific, and those that do typically take a traditional approach to cyber-security such as perimeter security or require centralized data aggregation. As highlighted by authors in [3], man-in-the-middle attacks have the ability relay falsified data that can deceive a centralized operator or intrusion detection system about the wind turbine's true operational status.

The present situation of cyber-security solutions for wind power systems has motivated the need for an intrusion detection system that is not reliant on centralized data acquisition. Presented in this paper is a fully distributed, model-based intrusion detection algorithm that has the ability to identify the presence of certain parameter manipulating cyber-attacks within a wind farm. The algorithm draws upon existing IDS schemes such as reputation scoring and collaborative nodes, but is unique in that it takes advantage of application-layer insight gained from understanding the interaction between wind speed and wind turbine control.

Properties from the Horns Rev wind farm in Denmark were used to help develop a credible simulation environment to test the algorithm. The algorithm was test under various simulation environments where cyber-attacks were emulated through parameter manipulations to turbine supervisory controllers. The number of corrupted turbines and the magnitude of parameter manipulation were varied independently and together to display the algorithm's performance under different attack attributes. Assessed using false positives and false negatives as metrics, the algorithm produced excellent results when the number of corrupted turbines was below 30 and the wind speed cut-in parameter manipulation was greater than 10 m/s. Based on these results, it is reasonable to conclude that the intrusion detection algorithm presented in this paper is worth future research and could potentially be used to help develop cyber-security solutions for wind power systems.

9. References

[1] United States. Dept. of Energy, *Wind power in America's future : 20% wind energy by 2030*. Mineola, N.Y.: Dover Publications, 2010.

[2] A. Hann, G. Manimaran, and P. Sauer, "Cyber-Physical Systems Security for Smart Grid." PSERC Publication 12-02, 2012.

[3] J. Yan, C.-C. Liu, and M. Govindarasu, "Cyber intrusion of wind farm SCADA system and its impact analysis," in *Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES*, 2011, pp. 1 –6.

[4] J. M. Moya, Á. Araujo, Z. Banković, J.-M. de Goyeneche, J. C. Vallejo, P. Malagón, D. Villanueva, D. Fraga, E. Romero, and J. Blesa, "Improving Security for SCADA Sensor Networks with Reputation Systems and Self-Organizing Maps," *Sensors*, vol. 9, no. 11, pp. 9380–9397, Nov. 2009.

[5] M. García-Sanz and C. H. Houpis, *Wind energy systems : control engineering design*. Boca Raton, FL: CRC Press, 2012.

[6] R. A. Jones and B. Horowitz, "A System-Aware Cyber Security architecture," *Systems Engineering*, vol. 15, no. 2, pp. 225–240, 2012.

[7] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *3rd IEEE Consumer Communications and Networking Conference, 2006. CCNC 2006*, 2006, vol. 1, pp. 640 – 644.

[8] G. Athanasios, "Intrusion Detection in Wireless Sensor Networks," Master's, Carnegie Melon University, 2008.

[9] P. Ebinger and N. Bissmeyer, "TEREC: Trust Evaluation and Reputation Exchange for Cooperative Intrusion Detection in MANETs," in *Communication Networks and Services Research Conference, 2009. CNSR '09. Seventh Annual*, 2009, pp. 378 –385.

[10] K. Gerrigagoitia, R. Uribeetxeberria, U. Zurutuza, and I. Arenaza, "Reputation-based Intrusion Detection System for wireless sensor networks," in *Complexity in Engineering (COMPENG), 2012*, 2012, pp. 1 –5.

[11] W. F. Young, J. E. Stamp, J. D. Dillinger, and M. A. Rumsey, "Communication Vulnerabilities and Mitigations in Wind Power SCADA Systems," in *Session 3B - Technology Performance Part 1*, Austin, TX, 2003.

[12] B. Zhu and S. Sastry, "SCADA-specific intrusion detection/prevention systems: a survey and taxonomy," in *Proceedings of the 1st Workshop on Secure Control Systems (SCS)*, 2010.

[13] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proceedings of the SCADA Security Scientific Symposium*, 2007, pp. 127–134.

[14] L. Y. Pao and K. E. Johnson, "A tutorial on the dynamics and control of wind turbines and wind farms," in *American Control Conference, 2009. ACC'09.*, 2009, pp. 2076–2089.

[15] "Alternative-Energy Wind Turbines," *Alternative-Energy*. .

[16] S. Miller, *Simulink - Wind Turbine Model*. 2009.

[17] J. F. Manwell, *Wind energy explained: theory, design and application*. Chichester ; New York: Wiley, 2002.

[18] *Wind power integration: connection and system operational aspects*. London: Institution of Engineering and Technology, 2007.

[19] C. Hasager, L. Rasmussen, A. Peña, L. Jensen, and P.-E. Réthoré, "Wind Farm Wake: The Horns Rev Photo Case," *Energies*, vol. 6, no. 2, pp. 696–716, Feb. 2013.

[20] R. J. Barthelmie, W. Schlez, A. Neubert, M. Heath, S. T. Frandsen, O. Rathmann, K. Hansen, E. Politis, J. Prospathopoulos, J. G. Schepers, K. Rados, D. Cabezón, and R. N. L. for S. E. W. E. D. Technical Univ. of Denmark, *Flow and wakes in large wind farms. Final report for UpWind WP8.* 2011.

[21] M. B. Christiansen, *Wind energy applications of synthetic aperture radar*, vol. 105. 2006.

[22] J. Kristoffersen, "The Horns Rev Wind Farm and The Operational Experience With The Wind Farm Main Controller," in *Copenhagen Offshore Wind*, Denmark, 2005, p. 9.

[23] Tech-wise, "Wind Resources at Horns Rev." Tech-wise, Dec-2002.