A Trust Modeling Framework with Application to Critical Infrastructures

Yujue Wang[1], Thoshitha Gamage[2] and Carl Hauser[3]

[1] Washington State University, Pullman WA, U.S.A.
ywang@eecs.wsu.edu
[2] Washington State University, Pullman WA, U.S.A.
tgamage@eecs.wsu.edu
[3] Washington State University, Pullman WA, U.S.A.
hauser@eecs.wsu.edu

Abstract

This paper connects the concept of trust with uncertainty and provides a definition of trust. This definition is used to propose a mathematical framework for trust modeling that is appropriate for critical infrastructure system applications. We provide two algorithms that are based on our framework with numerical simulations that sketches its effectiveness on a sample power grid application. Simulation results demonstrate that our algorithms are highly effective and accurate in making trust decisions.

1 Introduction

Trust is a relatively new area of research in computer science but has a rich and a mature basis in other research areas. Nearly every aspect of a person's social interaction involves some form of trust [13]. Thus, trust is as almost old as the human race. Trust research first appeared in sociology [11] and then unfolded over other areas including communications [15], economics [18], and political science [5]. In these areas, trust has been intensively studied in situations that require *collaboration* or information sharing under *uncertainty*.

Trust is especially becoming much more pronounced in research related to critical infrastructures. The main reason for this is that applications pertaining to critical infrastructures implicitly follows three tendencies. Firstly, modern systems are becoming more distributed geographically and are stretching across multiple domains (e.g. cyber-physical systems, social media, cloud computing, etc). Consequently, agents in these systems are more exposed to environments with more uncertainty. Secondly, distributed systems in mobile and ubiquitous environments demands much more frequent information sharing as well as smoother collaboration among heterogeneous agents. This again raises important trust issues. Thirdly, the traditional Trusted Third Party (TTP) approach is deemed neither scalable in expanding structures nor resistive to intentional malicious attacks.

Security in large multi-domain systems are also challenged by the lack of expressiveness, and suffer from numerous security implications. Trust, on the other hand, is more expressive than many traditional system properties, including security, due to its *abstract* and *multi-faceted* nature [14]. Moreover, trust models can cover the inherent uncertainties present in these complex distributed systems, which most traditional security models aren't capable of capturing.

Uncertainty has a great influence on performance in large complex distributed systems. Unfortunately this is something that is unavoidable because that the information that directly influences the reasoning (cause and effect) and decision making (rules) in these systems aren't always accurate nor complete [6]. Such information can come from different sources in different forms, which makes it fundamentally unsound to categorize or classify uncertainty into different types; there is only one kind of uncertainty – *the lack of knowledge* pertaining to the truth of a proposition [2].

In practice, however, making artificial distinctions on different types of uncertainties can make it easier to handle [16]. A common view is that uncertainties can be considered as either *aleatory* or *epistemic* [3]. Aleatory uncertainty comes from the natural and random variation of the system while epistemic uncertainty is due to the lack of knowledge. It is also possible to consider malicious behavior and attacks as another type of uncertainty since it leads to unpredictable behavior.

The overarching goal of this research is to build a mathematical framework for trust modeling that is suitable to the needs of modern critical infrastructure applications. The remainder of the this paper is organized as follows: Section 2 is a short introduction to the ontology of trust. Section 3 presents the proposed framework of trust modeling and Section 4 presents two algorithms that follow the proposed framework. We show some simulation results in Section 5 followed by conclusions in Section 6.

2 Definition of Trust

Definition 1 (Definition of Trust) : Trust is the subjective belief in the consistency between trustee's behavior and trustor's expectation under a given context in an environment with uncertainty.

At the conceptual level, trust is very *abstract* and *multifaceted* [14]. The sheer expressiveness of trust makes it possible to interpret and understand it in different ways. This however, has caused the lack of a widely-accepted canonical model of trust, either conceptually or mathematically. Besides, there is no trust model with a critical infrastructure emphasis within which uncertainty is very significant in terms of its protection and security. A definition of trust with implications to critical infrastructure applications has five essential aspects as follows. By taking these aspects in to consideration, a definition of trust for critical infrastructures is provided in Definition 1.

- **Uncertainty**: It only makes sense to talk about trust when facing uncertainty
- **Belief**: Trust is a psychological state i.e. trust is belief
- **Consistency**: Trust is a judgment on the consistency between trustee's behavior and trustor's expectation
- **Context**: A trustor only trusts a trustee under a specific context
- **Subjectivity**: Trust is subjective. Different trustors may make different trust decisions based on the same set of observation under same context

3 Trust Framework for Critical Infrastructure Systems

A generalized trust modeling problem can be depicted as follows: assume that there are n agents in a certain problem scope. Given a time window $\mathcal{T}$, timings t_i, $t_i \in \mathcal{T}$, and a potential trustee agent α, a trustor agent β can have an observation $x_{t_i}^{\alpha}$ of agent α's behavior on some context Ω. The series of observations $x^{\alpha}_{t_i}$ within $\mathcal{T}$ is denoted as X^{α}. Thus, the trust modeling problem can be defined as follows[1]:

Definition 2 (Trust Modeling Problem) Given a trustee agent α, a trustor agent β, and β's observation X on α within time window $\mathcal{T}$ on context Ω, β has to decide between the hypotheses:

- $\mathcal{H}_0$: trustee α is NOT trustworthy based on X;
- $\mathcal{H}_a$: trustee α is trustworthy based on X.

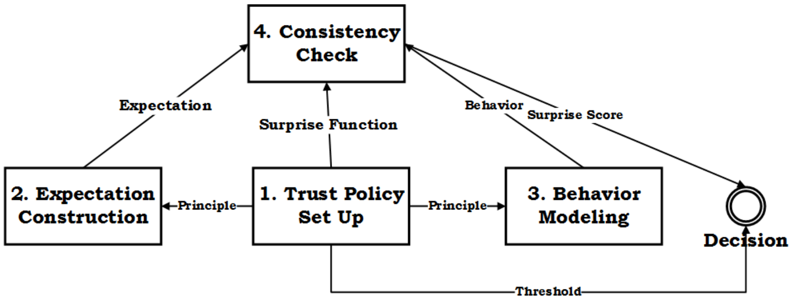

Figure 1: The Framework of Trust Modeling for Critical Infrastructures

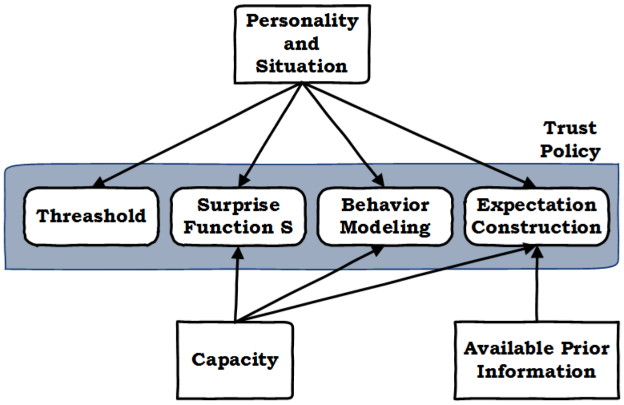

Figure 2: Trust Policy Setup

The aim of the proposed framework is to formulate the rules for the hypotheses testing in Definition 2. From Definition 1, trust modeling is essentially a measurement between the difference in trustor's expectation and the trustee's behavior. Specifically, this difference is called the *surprise* and the function to check the difference is termed the *surprise function*. With this, the two hypotheses in Definition 2 can be extended to include surprise as follows:

- $\mathcal{H}_0$: trustee α is NOT trustworthy based on X, if $\mathcal{S}(E, P) > \tau$;
- $\mathcal{H}_a$: trustee α is trustworthy based on X, if $\mathcal{S}(E, P) \leq \tau$

[1] When the trustee and time are obvious, we write X^{α} as X and $x^{\alpha}_{t_1}$ as x

where τ is a threshold determined by the trustor, E is the trustor's expectation, and P is the trustee's behavior which can be inferred from X. Both E and P can be either distributions (functions) or vectors. $\mathcal{S}$ is the surprise function.

The conceptual flow of the rules for this hypotheses testing, i.e., the framework of trust modeling, is illustrated in Figure 1. As shown, the whole trust modeling framework is composed of four phases: trust policy set up, expectation construction, behavior modeling, and consistency check. In the *trust policy set up* phase, trustors set up their principles for expectation constructions, behavior modeling and decision makings. In the *expectation construction* phase, trustors form their expectations. In *behavior modeling* phase, trustors model the behavior of trustee agents according to their observations X on these agents. In the final phase, trustors compute the surprise scores indicating the differences between their expectations and trustee agents' behavior models, compare these scores with the thresholds set up in trust policy set up phase, and conclude their final trust decisions.

3.1 Trust Policy Set Up

A trustor sets up his trust policy so that trustor can construct the expectation and the threshold for the hypotheses testing according to this policy. These policies should be made according to the trustor's context of *personality* which is his/her internal characteristic and *situation* which is his/her condition in the external environment.

Figure 2 shows the trust policy set up phase, which is made up of four factors: a threshold τ for the hypotheses testing, principle to construct expectation E, principle to model trustee's behavior P based on X, and surprise function $\mathcal{S}$. Practically, these factors are usually determined by three trustor attributes: *personality and situation*, *available prior information*, and *capability* to digest the information.

- **Personality** and **Situation**: Personality and situation are inherent to individual trustors, thus, they can influence all aspects of a trustor's trust policies. For example, trustors, who are less vulnerable to untrustworthy trustees, tend to be bold thus, the processes of making a trust decision is relatively easy. Conversely, trustors who are more vulnerable to untrustworthy trustees are usually cautious when making trust decisions. A super agent – an agent with a lot of interactions with a large pool of agents within the same context – maybe picky about trustees since they have more potential options and opinions. At the same time, their trust decisions might have greater impact on the whole system.

- **Available Information**: In addition to the behavior data X, a trustor may have some prior information about a trustee. This prior information determines how the trustor can construct his/her expectation about the trustee. Explicitly, the trustor may have:

  1. Sufficient prior information which means the trustor has very specific expectations on the trustee;

  2. Partial prior information signifying that the trustor knows the domain of the expectations for the trustee. The trustor may assign a weight to an expectation indicating the frequency of its occurrence. This forms a distribution of expectations;

  3. The trustor may have no prior information to construct his/her expectation(s) except some historical behavior data of trustee. So it would rely completely on the historical behavior data $X_{[1,t_i-1]}$ to construct his/her expectations or adopt expectation from other trusted peers.

- **Capacity**: The selection of function $\mathcal{S}$, trustee's behavior modeling, and the expectation construction are all dependent on trustor's computing capacity and capability to process information and data.

In practice, the process of setting up a trust policy is the most fundamental and difficult task in trust modeling. It has the greatest influence on the performance of the model and reflects trustor's subjectivity and flexibility. Trustors can construct their expectations and complete other trust modeling phases only after the trust policy is set up.

3.2 Expectation Construction

In this phase, the trustor should form his/her expectation according to some prior information such as trusted peers' endorsements. There are three cases for the available prior information:

1. The trustor has concrete expectation within the time window T so that the prior information will be $E = \{e\}$ (e stands for a concrete expectation);

2. The trustor has a set of possible expectations within time window T so that the prior information will be $E = \{e_1, e_2, \ldots, e_n, \ldots\}$ ($e_1, e_2, \ldots, e_n$ are concrete decisions). If the trustor assigns weight to every $e_i \in E$ indicating their frequencies of occurrences, this will form a distribution $f(e)$.

3. The trustor has no expectation and cannot obtain significant prior information about expectation. Thus, $E = \emptyset$. In order to complete the trust decision, the trustor has to construct his/her expectations E' based on historical observations X^- or adopt expectations E^a from other trusted relevant peers.

3.3 Behavior Modeling

In this phase, the trust model takes observation data X on the trustee's current behavior as input and outputs a model P describing trustee's behavior. For simplicity we directly use X as the model of trustee's behavior instead of constructing more complicated models.

3.4 Consistency Check

In this phase, the trustor's surprise function $\mathcal{S}$ determined in the policy set up phase takes the expectation E formed in the expectation construction phase and the behavior model P generated in the behavior modeling phase as inputs and outputs a score indicating the difference between trustor's expectation and the trustee's behavior. This value is then compared with the threshold τ which is predefined in the policy set up phase to decide whether trustor should reject $\mathcal{H}_0$.

3.5 Trust Modeling Framework

Context is implicitly fixed in certain trust modeling frameworks. A trustor begins with setting up his/her trust policy which can reflect his/her *subjectivity*. Then the modeling of expectation and trustee's behavior takes *uncertainty* into consideration. Consistency check procedure explicitly investigates the *consistency* between trustor's expectation and trustee's behavior. Trustor's *belief* of trust forms after trustor compares the surprise score and the threshold. So generally, our framework closely follows our definition and characteristics of trust.

4 Two Concrete Trust Models

In this section, we show three specific models corresponding to three cases for expectation construction. Firstly, we review the residue check method for bad data detection in power grid state estimation as a trust model complying with concrete expectation $E = e$ case. Then, we propose two concrete algorithms for trust modeling – the trust model with prior expectation (PE model) and the trust model with self-built expectation (SE model) – both of which closely follow the proposed framework. These two algorithms are corresponding to the case $E = \{e_1, e_2, \dots, e_n, \dots\}$ and $E = \emptyset$ respectively.

Power grid operation heavily relies on the measurement data, which defines the grid's state. These measurements are typically transmitted to a control center, which provides monitoring and control over the power system. State estimation, which is used to best estimate the power grid condition according to analysis of measurement data and power system models, is crucial to both this monitoring and control of the power grid [10]. For instance, consider a DC power state estimation based on a linearized AC power flow model [7]:

$$\mathbf{z}\!=\mathbf{H}\mathbf{x} + \mathbf{a} + \mathbf{e}$$

Here, $\mathbf{e}$ is the noise and $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \Sigma_e)$. And $\mathbf{a}$ is the malicious data injected by an adversary ($||\mathbf{a}||_0 \leq k$ means $\mathbf{a}$ is a vector with at most k non-zero entries). $\mathbf{z} \in R^m$ is the vector power flow measurements. $\mathbf{x} \in R^n$ is the system state.

Since the control center obtains the estimated state $\hat{\mathbf{x}}$, its expectation of the measurement $\mathbf{z}$ should be $E = \{\mathbf{H}\hat{\mathbf{x}}\}$. The surprise function $\mathcal{S}$ is the 2-norm of difference between $\mathbf{z}$ and $\mathbf{H}\hat{\mathbf{x}}$, which is $\mathbf{r} = ||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}||$. If $\mathbf{r} < \tau$, trustor rejects $\mathcal{H}_0$ and vice versa.

4.1 PE Model: Trust Model with Prior Expectation

We can make use of the Kolmogorov-Smirnov (KS) test to define the surprise function $\mathcal{S}$ for cases where the trustor has a distribution of expectations, $E : q = f(e_i), e_i \in E$. In $\mathcal{S}$, we consider the expectation as the reference probability distribution and X as the sample data. Essentially, the KS test is used to decide whether a sample is coming from a population with a specific distribution. Thus, the idea in the PE model is to test whether the behavior data of the trustee (sample data) follows the trustor's expectation (the specific given distribution of expectation). So in this model, trustor can make use of the observation data X on the trustee's behavior directly without modeling it.

KS test is based on the empirical cumulative distribution function (ECDF)[2]. Firstly, we reorganize the data of X in ascending order $X = \{x_1, x_2, \dots, x_n\}$. If $i < j$, then $x_i \leq x_j$, where $1 \leq i, j \leq n$. Then, by the definition of ECDF in [4], we can obtain the ECDF of X:

$$K_N(x_i) = \frac{n(i)}{N}$$

where $n(i)$ is the number of points less than x_i. The KS test is to measure the maximum distance between the cumulative distribution of expectations and the ECDF of the behavior data.

The surprise function (Kolmogorov-Smirnov test statistic) is defined as:

$$\mathcal{S} = \sup_{x_i} ||F(x_i) - K_N(x_i)||, \forall x_i \in X$$

[2] ECDF is the distribution function associated with the empirical measure of the sample

where F is the cumulative distribution of the expectation distribution which must be a continuous distribution and fully specified (i.e., the location, scale, and shape parameters cannot be estimated from the data). Thus the hypotheses can be rewritten as:

- $\mathcal{H}_0$: (Trustee is NOT trustworthy) $\mathcal{S}(E, X) = KS(F, K_N) > \tau$. The behavior data doesn't follow the expectation distribution;
- $\mathcal{H}_a$: (Trustee is trustworthy) $\mathcal{S}(E, X) = KS(F, K_N) \leq \tau$. The behavior data follows the expectation distribution.

4.2 SE Model: Trust Model with Self-built Expectation

In some situations, trustors in critical infrastructures don't have predefined expectations. Then, trustors can resort to constructing their own expectations based on historical observations. For the expectation established according to historical data, it is further assumed that they are highly correlated to the future behavior. More specifically, a trustor can predict the behavior of the trustee in some sense in accordance to the historical data.

Here, we apply the kernel functions [12] to construct the expectations. A kernel function is a weighting function used in non-parametric techniques that makes use of the data obtained to estimate the underlying distributions. Let $X_{t-1} = \{x_1, x_2, \ldots, x_n\}$ be the series of observation data within a time window T_{t-1} and assume that they are i.i.d. examples drawn according to a distribution $g(x)$. The Parzen-window estimate [17] of $g(x)$ is based on the n samples in X and is defined as:

$$\hat{g}(x) = \frac{1}{n} \sum_{i=1}^{n} \delta(x - x_i)$$

where $\delta(\cdot)$ is a kernel function with localized support and its exact form depends on n. The Gaussian kernel function is the most common kernel function because Gaussian function is smooth and hence the estimated distribution function $\hat{g}(x)$ also varies smoothly. Thus, $\hat{g}$ can be expressed as a Gaussian kernels with common variance σ^2:

$$\hat{g}(x) = \frac{1}{n(2\pi)^{\frac{d}{2}}\sigma^d} \sum_{i=1}^{n} \exp\{-\frac{||x - x_i||^2}{2\sigma^2}\}$$

where d is the dimensionality of the feature space.

Consider time window T_t right after T_{t-1}, which results in another time series of observation data X_t. Using the same methodology as above, we derive the Parzen-window estimator $\hat{h}(\tilde{x})$ based on X_t. So for the time instance t, $\hat{g}x$ is considered the expectation E and $\hat{h}(\tilde{x})$ as the behavior of trustee P. Then the question is how can we measurement the difference between this P and E which are both distributions?

For the surprise function, we measure the difference between two distributions $\hat{g}(x)$ and $\hat{h}(\tilde{x})$ with relative entropy (Kullback-Leibler (KL) divergence) [8]:

$$\mathcal{S} = KL(\hat{g}(x), \hat{h}(\tilde{x})) = \int_{-\infty}^{\infty} \ln(\frac{\hat{g}(x)}{\hat{h}(\tilde{x})})\hat{g}(x)dx$$

Thus. the hypotheses can be adapted as:

- $\mathcal{H}_0$: (Trustee is NOT trustworthy) $\mathcal{S}(E, X) = KL(\hat{g}(x), \hat{h}(\tilde{x})) > \tau$;
- $\mathcal{H}_a$: (Trustee is trustworthy) $\mathcal{S}(E, X) = KL(\hat{g}(x), \hat{h}(\tilde{x})) \leq \tau$.

5 Numerical Results

In this section, we test our models on data sets generated by **GridSim** [1]. GridSim is a power grid simulation software which generates simulated PMU data streams at a rate of 30 samples/sec which includes time, voltage, current, and phase angle measurements. The data is assumed noise free as they are generated by solving highly accurate power system differential equations. For experimental purposes, we manipulate the generated data streams by injecting random vectors to them and then use the original data as data from "trustworthy agents" and manipulated data as data from "untrustworthy agents".

We intercepted 5 minutes of GridSim data and extracted 101 streams of voltage readings between different pairs of buses in Kundur's 2-area system [9]. There were 3 observable oscillations in this 5 minute window. For comparison and performance measurement purposes, we separated the data frames in both trustworthy and untrustworthy data streams into data frames with oscillation and data frames without oscillation and ran our algorithms on both of these kinds of frames.

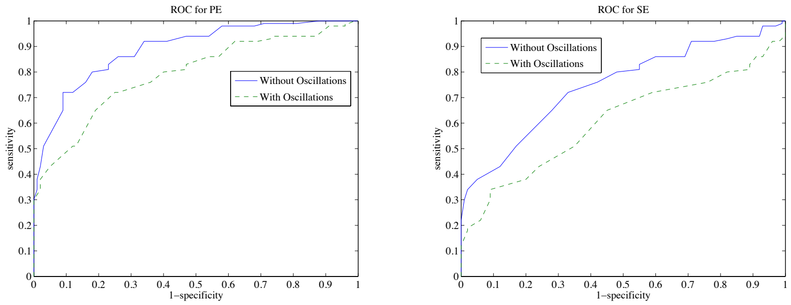

Figure 3: ROC Curves of PE Model (left) and SE Model (right)

We tested both of the PE model and SE model on these two types of data frames (with and without oscillation) and measured the *sensitivity* and *specificity* of our algorithms. In statistics, sensitivity and specificity are measures of performance of a binary classification test. Sensitivity measures the proportion of actual positives which are correctly identified as such and specificity measures the proportion of negatives which are correctly identified. In other words, *sensitivity* = (1− type I error) [3] and *specificity* = (1− type II error) [4].

5.1 Simulations of the PE Model

Figure 3 illustrates a ROC curves for the PE model on both data frames with oscillation and data streams without oscillation. The ROC curve is a comprehensive indicator of the performance that is created by plotting the *sensitivity* vs. (1−*specificity*) (i.e., type II error) at various thresholds of τ. Since all the data streams are from the same bus system, it is expected that their behavior to be highly related. We arbitrarily pick one of these data streams as the a trustworthy *reference data stream* to construct the expectations for all other streams.

To test the accuracy of the models, we manipulate the other 100 data streams without oscillation data frames and inject random "attack vectors" to emulate "data streams from untrustworthy PMUs". Then we label the original data streams as "trustworthy data streams"

[3] Type I error occurs when the null hypothesis ($\mathcal{H}_0$) is true, but is rejected

[4] Type II error occurs when $\mathcal{H}_0$ is false but is accepted

and manipulated ones as "untrustworthy data streams". We perform a KS test on both the trustworthy data streams and the untrustworthy data streams to get the KS statistic values respectively. We make different trust decisions on these data streams as the threshold τ varies between $[0, 1]$. The same procedure is applied to the 101 data streams with oscillations to obtain its respective ROC curve.

The area under the curve (AUC) of ROC is a number indicating the comprehensive performance of the respective model. Higher AUC number means greater accuracy on trust decisions. ROC in Figure 3 without oscillation is 0.8894, which indicates that for the PE model, the probability that a randomly chosen trustworthy data stream gaining higher KS score than a randomly chosen untrustworthy data stream is 0.8894. AUC of ROC with oscillation is 0.7908.

5.2 Simulation on the SE Model

We draw a ROC for the SE model by following the same procedure and by injecting a random attack vector as in the PE model above. This is illustrated in Figure 3. Here, we use a one second time window and construct distributions according to Parzen-window for every time window. Then we apply the surprise function to measure the relative entropy between every two consecutive time windows and normalize the data. The AUC of ROC for SE model without oscillations is 0.7474 and AUC of ROC with oscillations is 0.6072. What this means is that the probability that a randomly chosen untrustworthy data stream gaining higher score of relative entropy than a randomly chosen untrustworthy data stream is 0.8894 for data streams without oscillation. The same comparison for data streams with oscillations is 0.6072.

As ROC curves in Figure 3 show, both PE and SE models work better on data without oscillation than on data with oscillation. Our explanation for this is that both the PE model and SE model are based on some significant assumptions and oscillations break these assumptions. For PE model, it is implicitly assumed that all other data streams follow the data stream that is chosen as the reference expectation. SE model assumes that data points from the same data stream follow the same distribution. When oscillation occurs, the distributions change in unpredictable ways and violate these assumptions.

6 Conclusion and Future Work

In this paper, we proposed a definition of trust that is suitable for applications in critical infrastructures. There are five factors in this definition: *uncertainty*, *belief*, *consistency*, *context* and *subjectivity*. Based on these factors, we build a mathematical framework that is capable of adapting to situations with different trust policies and prior information. Specifically, we proposed a model that fits cases with a set of known expectations (PE model) and another model which is suitable to applications without foregone expectations available (SE model). Simulation results show: (1) that the trustors have the flexibility to choose a model that best fits the trust decisions they desire; and (2) that both the SE and PE models work well on both data with and without oscillations. Some of the planned future work includes further exploration on trust policy construction and advanced techniques to develop expectations from prior information.

References

[1] D. Anderson, Chuanlin Zhao, C.H. Hauser, V. Venkatasubramanian, D.E. Bakken, and A. Bose. A Virtual Smart Grid: Real-Time Simulation for Smart Grid Control and Communications Design. *IEEE Power and Energy Magazine*, 10(1):49–57, 2012.

[2] George Apostolakis. The Concept of Probability in Safety Assessments of Technological Systems. *Science*, 250(4986):1359–1364, 1990.

[3] Stephen C. Hora. Aleatory and Epistemic Uncertainty in Probability Elicitation with an Example from Hazardous Waste Management. *Reliability Engineering and System Safety*, 54(2-3):217–223, Nov.-Dec. 1999.

[4] R. G. Laha Indra Mohan Chakravarti and J. Roy. *Handbook of Methods of Applied Statistics*, volume 1. John Wiley and Sons Inc., 1967.

[5] J. Jack and Donald P. Green. Presidential Leadership and the Resurgence of Trust in Government. *British Journal of Political Science*, 16:143–53, 1986.

[6] Daphne Koller and Avi Pfeffer. Probabilistic Frame-based Systems. In *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, AAAI '98/IAAI '98, pages 580–587, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

[7] O. Kosut, Liyan Jia, R.J. Thomas, and Lang Tong. On Malicious Data Attacks on Power System State Estimation. In *Proceedings of 45th International Universities Power Engineering Conference (UPEC)*, pages 1–6, 2010.

[8] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[9] Prabha Kundur. *Power System Stability and Control*. McGraw-Hill Professional, 1994.

[10] Yao Liu, Peng Ning, and Michael K. Reiter. False Data Injection Attacks Against State Estimation in Electric Power Grids. In *Proceedings of the 16th ACM Cinproceedings on Computer and Communications Security*, CCS '09, pages 21–32, New York, NY, USA, 2009. ACM.

[11] Niklas Luhmann. *Trust and Power*. John Wiley and Sons Inc, 1969.

[12] Emanuel Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[13] Ye Diana Wang and Henry H. Emurian. An Overview of Online Trust: Concepts, Elements, and Implications. *Computers in Human Behavior*, 21(1):105–125, 2005.

[14] Ye Diana Wang and Henry H. Emurian. An Overview of Online Trust: Concepts, Elements, and Implications. *Computers in Human Behavior*, 21(1):105–125, 2005.

[15] Lawrence R. Wheeless and Janis Grotz. The Measurement of Trust and its Relationship to Self-disclosure. *Human Communication Research*, 3(3):250–257, 1977.

[16] Robert L. Winkler. Uncertainty in Probabilistic Risk Assessment. *Reliability Engineering and System Safety*, 54(2-3):127–132, Nov.-Dec. 1999.

[17] Dit-Yan Yeung and C. Chow. Parzen-Window Network Intrusion Detectors. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 4, pages 385–388, 2002.

[18] Louise C. Young and Ian F. Wilkinson. The Role of Trust and Co-operation in Marketing Channels: A Preliminary Study. *European Journal of Marketing*, 23(2):109–122, 1989.