**SANDIA REPORT**
SAND2012-10177
Unlimited Release
Printed November 2012

# Proactive Defense for Evolving Cyber Threats

Richard Colbaugh and Kristin Glass

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation,
a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's
National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

**NOTICE:** This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

|  |  |
|---|---|
| Telephone: | (865) 576-8401 |
| Facsimile: | (865) 576-5728 |
| E-Mail: | reports@adonis.osti.gov |
| Online ordering: | http://www.osti.gov/bridge |

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

|  |  |
|---|---|
| Telephone: | (800) 553-6847 |
| Facsimile: | (703) 605-6900 |
| E-Mail: | orders@ntis.fedworld.gov |
| Online order: | http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online |

SAND2012-10177
Unlimited Release
Printed November 2012

# Proactive Defense for Evolving Cyber Threats

Richard Colbaugh
Analytics and Cryptography

Kristin Glass
Cyber Research and Education

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MSXXXX

## Abstract

There is great interest to develop proactive methods of cyber defense, in which future attack strategies are anticipated and these insights are incorporated into defense designs; however, little has been done to place this ambitious objective on a sound scientific foundation. Indeed, even fundamental issues associated with how the "arms race" between attackers and defenders actually leads to predictability in attacker activity, or how to effectively and scalably detect this predictability in the relational/temporal data streams generated by attacker/defender adaptation, haven't been resolved. This LDRD project addressed many of these challenges and the results are briefly summarized here.

We have characterized the predictability of attacker/defender coevolution and have leveraged our findings to create a framework for designing proactive defenses for large (organizational) networks. More specifically, this project applied rigorous predictability-based analytics to two central and complementary aspects of the network defense problem – attack strategies of the adversaries and vulnerabilities of the defenders' systems – and used the results to develop a scientifically-grounded, practically-implementable methodology for designing proactive cyber defense systems. Briefly, predictive analysis of attack strategies involved first conducting predictability assessments to characterize attacker adaptation patterns in given domains, and then used these patterns to "train" adaptive defense systems capable of providing robust performance against both current and (near) future threats.

The problem of identifying and prioritizing defender system vulnerabilities was addressed using statistical and machine learning to analyze a broad range of data (e.g., cyber, social media) on recently detected system vulnerabilities to "learn" classifiers that predict how likely it is that, and how soon, new vulnerabilities will be exploited. A variety of cyber threat case studies were developed and investigated throughout the project, one selected from the cyber security research community and one that is more comprehensive and of higher priority to SNL and to external national security partners.

A sample of research results and application of this methodology are included in this report (as a series of peer-reviewed publications). For ease of reference the title and SAND number are included below.

# CONTENTS

# Early warning analysis for social diffusion events

Richard Colbaugh[1] and Kristin Glass[2]

[1] Analytics and Cryptography Department, Sandia National Laboratories, Albuquerque, USA
[2]Cyber Research and Education Department, Sandia National Laboratories, Albuquerque, USA

RC: colbaugh@comcast.net
KG: kglass609@comcast.net (corresponding author)

**Abstract --** There is considerable interest in developing predictive capabilities for social diffusion processes, for instance to permit early identification of emerging contentious situations, rapid detection of disease outbreaks, or accurate forecasting of the ultimate reach of potentially "viral" ideas or behaviors. This paper proposes a new approach to this predictive analytics problem, in which analysis of meso-scale network dynamics is leveraged to generate useful predictions for complex social phenomena. We begin by deriving a stochastic hybrid dynamical systems (S-HDS) model for diffusion processes taking place over social networks with realistic topologies; this modeling approach is inspired by recent work in biology demonstrating that S-HDS offer a useful mathematical formalism with which to represent complex, multi-scale biological network dynamics. We then perform formal stochastic reachability analysis with this S-HDS model and conclude that the outcomes of social diffusion processes may depend crucially upon the way the early dynamics of the process interacts with the underlying network's *community structure* and *core-periphery structure*. This theoretical finding provides the foundations for developing a machine learning algorithm that enables accurate early warning analysis for social diffusion events. The utility of the warning algorithm, and the power of network-based predictive metrics, are demonstrated through an empirical investigation of the propagation of political "memes" over social media networks. Additionally, we illustrate the potential of the approach for security informatics applications through case studies involving early warning analysis of large-scale protests events and politically-motivated cyber attacks.

**Keywords:** social dynamics, predictive analysis, early warning, protest and mobilization, cyber security, security informatics.

## 1. Introduction

Understanding the way information, behaviors, innovations, and diseases propagate over social networks is of great importance in a wide variety of domains [e.g., 1-4], including national security [e.g., 5-13]. Of particular interest are predictive capabilities for social diffusion, for instance to enable early warning concerning the emergence of a violent conflict or outbreak of an epidemic. As a consequence, vast resources are devoted to the task of predicting the outcomes of diffusion processes, but the quality of such predictions is often poor. It is tempting to conclude that the problem is one of insufficient information. Clearly diffusion phenomena which "go viral" are qualitatively different from those that don't or they wouldn't be so dominant, the conventional wisdom goes, so in order to make good predictions we must collect enough data to allow these crucial differences to be identified.

Recent research calls into question this intuitively plausible premise and, indeed, indicates that intuition can be an unreliable guide to constructing successful prediction methods. For example, studies of the predictability of popular culture indicate that the *intrinsic* attributes commonly believed to be important when assessing the likelihood of adoption of cultural products, such as the quality of the product itself, do not possess much predictive power [14-16]. This research offers evidence that, when individuals are influenced by the actions of others, it may not be possible to obtain reliable predictions using methods which focus on intrinsics alone; instead, it may be necessary to incorporate aspects of *social influence* into the prediction process. Very recently a handful of investigations have shown the value of considering even simple and indirect measures of social influence, such as early social media “buzz”, when forming predictions. This work has produced useful prediction algorithms for an array of social phenomena, including markets [16-21], political and social movements [17,22], mobilization and protest behavior [23,24], epidemics [17,25], social media dynamics [26,27], and the evolution of cyber threats [28].

Recognizing the importance of accounting for social influence, this paper proposes a predictive methodology which explicitly considers the way individuals influence one another *through their social networks*. It is expected that prediction algorithms which are based, in part, on network dynamics metrics will outperform existing methods and be applicable to a wider range of diffusion systems. We begin by developing a stochastic hybrid dynamical systems (S-HDS) model for diffusion processes taking place over social networks with realistic topologies. This modeling approach is inspired by recent work in biology demonstrating that S-HDS offer a useful mathematical formalism with which to represent multi-scale biological network dynamics [29-33]. An S-HDS is a feedback interconnection of a discrete-state stochastic process, such as a Markov chain, with a family of continuous-state stochastic dynamical systems [34]. Combining discrete and continuous dynamics in this way provides a rigorous, expressive, and computationally-tractable framework for modeling the dynamics of the complex, highly-evolved networks that are ubiquitous in biological systems [35], and we show in this paper that the S-HDS framework is also well-suited to the task of modeling the network dynamics which underlie social diffusion.

With the S-HDS model in hand, we then perform formal stochastic reachability analysis and conclude that the outcomes of social diffusion processes may depend crucially upon the way the early dynamics of the process propagates with respect to the underlying network’s 1.) *community structure*, that is, densely connected groupings of individuals which have only relatively few links to other groups [36], and 2.) *core-periphery structure*, reflecting the presence of a small group of “core” individuals that are densely connected to each other and are also close to the remainder of the network [36]. This theoretical finding leads to the identification of novel metrics for the community and core-periphery dynamics which should be useful early indicators of which diffusion events will propagate widely, ultimately affecting a substantial portion of the population of interest, and which will not. Prediction is accomplished with a machine learning algorithm [37] which is based, in part, on these network dynamics metrics.

The paper makes three main contributions. First, we present a new S-HDS-based framework for modeling social diffusion on networks of real-world scale and complexity, enabling these dynamics to be appropriately represented as multi-scale phenomena. Second, we formulate predictive analysis problems as questions concerning the reachability of diffusion events, and present a novel “altitude function” method for assessing reachability *without simulating system trajectories*. The altitude function technique is both mathematically rigorous and computationally tractable, thereby permitting the derivation of provably-correct assessments for complex, large-scale systems. Third, the S-HDS model and altitude function analytics are used to characterize the importance of *meso-scale* network features, specifically network community and core-periphery structures, for understanding diffusion processes and predicting their fates. This characterization, in turn, forms the foundation for developing a new machine learning-based classification algorithm which employs these network dynamics features for accurate early warning analysis. Additionally, we evaluate the efficacy of this early warning algorithm through three empirical case studies investigating: 1.) the propagation of political “memes” [38] over social media networks, 2.) warning analysis for large-scale mobilization and protest events, and 3.) early warning for politically-motivated cyber attacks. These empirical studies illustrate the effectiveness of the proposed early warning methodology and demonstrate the significant predictive power of meso-scale network metrics for social diffusion processes. Moreover, the results indicate that the proposed algorithm provides a readily-implementable Web-based tool for early warning analysis for important classes of security-relevant diffusion events.

## 2. Early Warning Methodology

This section begins by defining the class of early warning problems of interest, then presents a brief, intuitive summary of the proposed social diffusion modeling and predictive analysis procedure, and finally describes the early warning indicators identified through this analytic procedure and the warning algorithm that is derived based on these results. A detailed mathematical presentation of the modeling and analysis methods is provided in Appendices One and Two.

### 2.1 Problem Formulation

The objective of this paper is to develop a scientifically-rigorous, practically-implementable methodology for performing early warning analysis for social diffusion events. Roughly speaking, we suppose that some "triggering event" has taken place or contentious issue is emerging, and we wish to determine, as early as possible, whether this event or issue will ultimately generate a large, self-sustaining reaction, involving the diffusion of discussions and actions through a substantial segment of a population, or will instead quickly dissipate. An illustrative example of the basic idea is provided by the contrasting reactions to 1.) the publication in September 2005 of cartoons depicting Mohammad in the Danish newspaper *Jyllands-Posten*, and 2.) the lecture given by Pope Benedict XVI in September 2006 quoting controversial material concerning Islam. While each event appeared at the outset to have the potential to trigger significant protests, the "Danish cartoons" incident ultimately led to substantial Muslim mobilization, including massive protests and considerable violence, while outrage triggered by the pope lecture quickly subsided with essentially no violence. It would obviously be very useful to have the capability to distinguish these two types of reaction as early in the event lifecycle as possible.

In order to state the early warning problem more precisely, we make a few assumptions:

- We suppose that the triggering event or emerging situation is given. Note that this is often the case in national security settings, and that additionally there exist techniques for *discovering* such events or issues in an automated or semi-automated manner [e.g., 24,27].
- It is assumed that data are available which provide a view of the early reaction of a relevant population to the trigger or issue of interest. These data can be only indirectly related to the event; for example, in this paper the primary data source is social media discussions (e.g., blog posts) while the events of interest are "real-world" activities such as protests.
- It is expected that the "customer" for the analysis provides at least qualitative definitions of the population of interest and the scale of reaction for which a warning is desired. Thus, for instance, in the example above, it might be of interest to anticipate Muslim reaction to the triggering incident, and to obtain a warning alert if the reaction is likely to eventually include self-sustaining, violent protests.

We formulate the early warning problem as a classification task. More specifically, given a triggering incident, one or more information sources which reflect (perhaps indirectly) the reaction to this trigger by a population of interest (e.g., social media discussions, intelligence reporting), and a definition for what constitutes an "alarming" reaction, the goal is to design a classifier which accurately predicts, as early as possible, whether or not reaction to the event will ultimately become alarming. Note that a more mathematically precise statement of this warning problem is given in Appendix Two. Observe that this type of warning analysis is both important in applications and “easier” to accomplish than more standard prediction or forecasting goals. Consider, as a familiar non-security example, the case of movie success. It is shown in [14-16] that it is likely to be impossible to predict movie revenues, even very roughly, based on the intrinsic information available concerning the movie ex ante (e.g., personnel, genre, critic reviews). However, we have demonstrated that it *is* possible to identify early indicators of movie success, such as temporal patterns in pre-release “buzz”, and to use these indicators to accurately predict ultimate box office revenues [39]. Recent research indicates that this result holds more generally, so that it may be more scientifically-sensible in many domains to pursue early warning rather than ex ante prediction goals [14-28].

### 2.2 S-HDS Social Diffusion Model

In social diffusion, individuals are affected by what others do. This is easy to visualize in the case of disease transmission, with infections being passed from person to person. Information, innovations, behaviors, and so on can also propagate through a population, as individuals become aware of a new piece of information or an activity and are persuaded of its relevance and utility through their social and information networks. The dynamics of social diffusion can therefore depend upon the topological features of the pertinent networks, such as the presence of highly connected blogs in a social media network (see, e.g., [4]). Indeed, social scientists have developed extensive theories explaining the role of social networks in the dynamics of social diffusion and mobilization (see the books [2-4] and the references therein, and also Appendix One, for discussions of this work). This dependence suggests that, in order to understand the predictability of social diffusion phenomena and in particular to identify features which possess predictive power, it is necessary to conduct the analysis using social and information network models with realistic topologies.

The social diffusion models examined in this study possess networks with three topological properties that are ubiquitous in real-world social and information networks and which have the potential to impact diffusion dynamics [36]:

- *transitivity* – the property that the network neighbors of a given individual have a heightened probability of being connected to one another;
- *community structure* – the presence of densely connected groupings of individuals which have only relatively few links to other groups;
- *core-periphery structure* – the presence of a small group of “core” individuals which are densely connected to each other and are also close to the other individuals in the network.

Additionally, we permit our network models to possess *right-skewed degree distributions,* in which most individuals have only a few network neighbors while a few individuals have a great many neighbors, as such networks are common in online settings. The manner in which the communities and the core-periphery are arranged will be said to define the network’s *meso-scale* structure. For convenience of exposition, the subsets of individuals specified by a partitioning of the network into communities and into a core and periphery will sometimes be referred to as the *partition elements*, and the collection of these (community and core-periphery) subsets will be called the *network partition*.

In order to deal effectively with networks possessing realistic topologies, and in particular to represent and analyze the way social dynamics is affected by the meso-scale structure, we model social diffusion in a manner which explicitly separates the individual, or “micro”, dynamics from the collective dynamics. More specifically, we adopt a multi-scale modeling framework consisting of three network scales:

- a *micro-scale*, for modeling the behavior of individuals;
- a *meso-scale*, which represents the interaction dynamics of individuals within the same network partition element (community or core/periphery);
- a *macro-scale*, which characterizes the interaction between partition elements.

The micro-scale quantifies the way individuals combine their own inherent preferences or attributes with the influences of others to arrive at their chosen courses of action. It is shown in Appendix One that separating the micro-scale dynamics from the meso- and macro-scale activity permits the dependence of this decision-making process on the social network to be characterized in a surprisingly straightforward way. The meso- and macro-scale components of the proposed modeling framework together quantify the way the decision-making processes of individuals interact to produce collective behavior at the population level. The role of the meso-scale model is to quantify and illuminate the manner in which behaviors *within* each network partition element (communities, core or periphery), while the macro-scale model captures the interactions *between* these elements. The primary assumptions are that interactions between individuals belonging to the same network partition element can be modeled more simply than those between individuals from distinct partition elements, and that the latter interactions are constrained by the "meta-network" which defines the dependencies between the partition elements.

This perspective offers a number of advantages. For example, at the micro-scale it is possible to unify behaviors which appear different phenomenologically but actually possess equivalent dynamics. We show in Appendix One that the social dynamics associated with classical "utility-maximizing" behavior and those arising from individuals attempting to infer information by observing the actions of others can be represented with the *same* micro-scale model. Additionally, separating the individual and collective dynamics supports efficient and flexible model building and simplifies the process of estimating model components from empirical data [39]. Dividing the collective dynamics into meso- and macro-scales also provides a mathematically-tractable, sociologically-sensible means of representing complex social network dynamics. For instance, because network communities are topological structures corresponding to localized social settings in the real world, determined by workplace, family, physical neighborhood, and so on, it is natural both mathematically and sociologically to model the interactions of individuals *within* communities as qualitatively different (e.g., more frequent and homogeneous) than those *between* communities.

Developing a mathematically-rigorous, expressive, scalable, and computationally-tractable framework within which multi-scale social network diffusion models can be constructed is, of course, a challenging undertaking. Recent work in systems biology has demonstrated that stochastic hybrid dynamical systems (S-HDS) provide a useful mathematical formalism with which to represent biological network dynamics that possess multiple temporal and spatial scales [29-33]. An S-HDS is a feedback interconnection of a discrete-state stochastic process, such as a Markov chain, with a family of continuous-state stochastic dynamical systems [34]. Thus the discrete system dynamics depends on the continuous system state, perhaps because different regions of the continuous state space are associated with different matrices of Markov state transition probabilities, and the particular continuous system which is "active" at a given time depends on the discrete system state. Combining discrete and continuous dynamics in this way provides an effective framework for modeling the dynamics of the complex, highly-evolved networks that are ubiquitous in biological systems [35]. For example, the rigorous yet tractable integration of switching behavior with continuous dynamics enabled by the S-HDS model allows accurate and efficient representation of biological phenomena evolving over disparate temporal scales [29-31] and spatial scales [32,33].

Inspired by this work, in this paper we apply the S-HDS framework to social diffusion dynamics evolving over multiple *network* scales. Appendix One provides a detailed discussion of the proposed S-HDS social diffusion model and demonstrates the effectiveness with which this formalism captures multi-scale network dynamics. As an intuitive illustration of the way S-HDS enable complex network phenomena to be efficiently represented, consider the task of modeling diffusion on a network that possesses community structure. As shown in Figure 1, this diffusion consists of two components: 1.) *intra-community dynamics*, involving frequent interactions between individuals within the same community and the resulting gradual change in the concentrations of "infected" (red) individuals, and 2.) *inter-community dynamics*, in which the "infection" jumps from one community to another, for instance because an infected individual "visits" a new community. S-HDS models offer a natural framework for representing these dynamics, with the S-HDS continuous system modeling the intra-community dynamics (e.g., via stochastic differential equations), the discrete system capturing the inter-community dynamics (e.g., using a Markov chain), and the interplay between these dynamics being represented by the S-HDS feedback structure. A detailed description of the manner in which S-HDS models can be used to capture social diffusion on networks with realistic topologies is given in Appendix One.

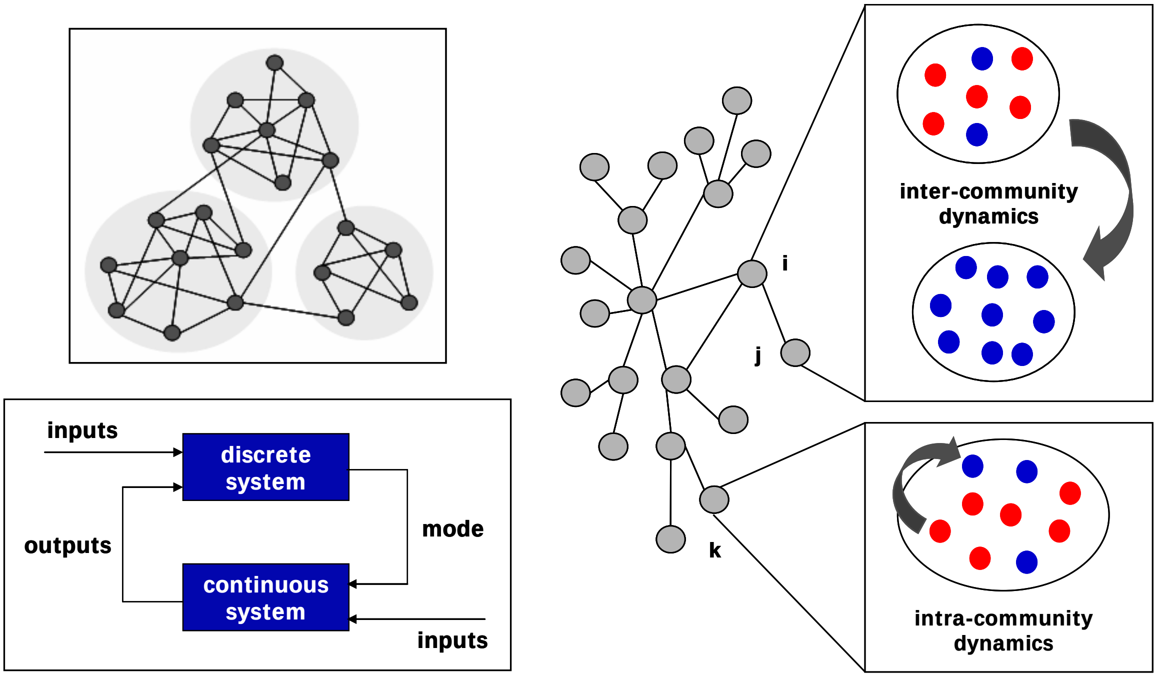


**Figure 1.** Modeling diffusion on networks with community structure via S-HDS. The cartoon at top left depicts a network with three communities. The cartoon at right illustrates diffusion *within* a community k and *between* communities i and j. The schematic at bottom left shows the basic S-HDS feedback structure; the discrete and continuous systems in this framework model the inter-community and intra-community diffusion dynamics, respectively.

### 2.3 Predictability Assessment

One hallmark of social diffusion processes is their ostensible unpredictability: phenomena from hits and flops in cultural markets to financial system bubbles and crashes to political upheavals appear resistant to predictive analysis (although there is no shortage of ex post explanations for their occurrence!). It is not difficult to gain an intuitive understanding of the basis for this unpredictability. Individual preferences and susceptibilities are mapped to collective outcomes through an intricate, dynamical process in which people react individually to an environment consisting largely of others who are reacting likewise. Because of this feedback dynamics, the collective outcome can be quite different from one implied by a simple aggregation of individual preferences; standard prediction methods, which typically are based on such aggregation ideas, do not capture these dynamics and therefore are often unsuccessful.

This section provides a brief, intuitive introduction to a systematic approach to assessing the predictability of social diffusion processes and identifying process observables which have exploitable predictive power (see Appendix Two, and also [17,39], for the mathematical details). Consider a simple model for product adoption, in which individuals combine their own preferences and opinions regarding the available options with their observations of the actions of others to arrive at their decisions about which product to adopt. As discussed above, it can be quite difficult to determine which characteristics of the process by which adoption decisions propagate, if any, are predictive of things like the speed or ultimate reach of the propagation [15-17]. In Appendix Two we propose a mathematically rigorous approach to predictability assessment which, among other things, permits identification of features of social dynamics which should have predictive power. We now summarize this assessment methodology.

The basic idea behind the proposed approach to predictability analysis is simple and natural: we assess predictability by answering questions about the reachability of diffusion events. To obtain a mathematical formulation of this strategy, the behavior about which predictions are to be made is used to define the system *state space subsets of interest* (SSI), while the particular set of candidate measurables under consideration allows identification of the *candidate starting set* (CSS), that is, the set of states and system parameter values which represent initializations that are consistent with, and equivalent under, the presumed observational capability. As a simple example, consider an online market with two products, A and B, and suppose the system state variables consist of the current market share for A, ms(A), and the rate of change of this market share, r(A) (ms(B) and r(B) are not independent state variables because ms(A) + ms(B) = 1 and r(A) + r(B) = 0); let the parameters be the advertising budgets for the products, bud(A) and bud(B). The producer of item A might find it useful to define the SSI to reflect market share dominance by A, that is, the subset of the two-dimensional state space where ms(A) exceeds a specified threshold (and r(A) can take any value). If only market share and advertising budgets can be measured then the CSS is the one-dimensional subset of state-parameter space consisting of the initial magnitudes for ms(A), bud(A), and bud(B), with r(A) unspecified (the one-dimensional "uncertainty" in the CSS reflects the fact that r(A) is not measurable).

Roughly speaking, the proposed approach to predictability assessment involves determining how probable it is to reach the SSI from a CSS and deciding if these reachability properties are compatible with the prediction goals. If a system's reachability characteristics are incompatible with the given prediction question – if, say, "hit" and "flop" states in the online market example are both fairly likely to be reached from the CSS – then the situation is deemed unpredictable. This setup permits the identification of candidate predictive measurables: these are the measurable states and/or parameters for which predictability is most sensitive (see Appendix Two). Continuing with the online market example, if trajectories with positive early market share rates r(A) are much more likely to yield market share dominance for A than are trajectories with negative early r(A), then the situation is unpredictable (because the outcome depends sensitively on r(A) and this quantity is not measured). Moreover, this analysis suggests that market share rate is likely to possess predictive power, so it may be possible to increase predictability by adding the capacity to measure this quantity.

A key element of this approach to predictability assessment is the proposed method of estimating the probability of reaching the SSI from a CSS. Note that in a typical assessment such estimates must be computed for several CSS in order to adequately explore the space of candidate predictive features, so that it is crucial to perform these estimates efficiently. In Appendix Two we develop an "altitude function" approach to this reachability problem, in which we seek a scalar function of the system state that permits conclusions to be made regarding reachability *without computing system trajectories*. We refer to these as altitude functions to provide an intuitive sense of their analytic role: if some measure of "altitude" is low on the CSS and high on an SSI, and if the expected rate of change of altitude along system trajectories is nonincreasing, then it is unlikely for trajectories to reach this SSI from the CSS. Moreover, the difference in altitudes between the CSS and SSI gives a measure of the probability of reaching the latter from the former. Because the reach probability is computed for *sets* of states without simulating system trajectories, the altitude function method offers an extremely efficient way to explore the space of candidate predictive features.

We have applied the predictability assessment methodology summarized above to the social diffusion prediction problem, and we now summarize the main conclusions of this study; a more complete discussion of this investigation is given in Appendix Two. The analysis uses the mathematically rigorous predictability assessment procedure summarized above, in combination with empirically-grounded S-HDS models for social dynamics, to characterize the predictability of social diffusion on networks with realistic degree distributions, transitivity, community structure, and core-periphery structure. The main finding of the study, from the perspective of the present paper, is that the predictability of these diffusion models depends crucially upon social and information network topology, and in particular on the community and core-periphery structures of these networks.

In order to describe these theoretical results more quantitatively and leverage them for prediction, it is necessary to specify mathematical definitions for network communities and core-periphery structure. There exist several qualitative and quantitative definitions for the concept of community structure in networks. Here we adopt the *modularity-based* definition proposed in [40], whereby a good partitioning of a network's vertices into communities is one for which the number of edges between putative communities is smaller than would be expected in a random partitioning. To be concrete, a modularity-based partitioning of a network into two communities maximizes the modularity Q, defined as

$$Q = s^T B\, s / 4m,$$

where m is the total number of edges in the network, the partition is specified with the elements of vector s by setting $s_i = 1$ if vertex i belongs to community 1 and $s_i = -1$ if it belongs to community 2, and the matrix B has elements $B_{ij} = A_{ij} - k_i k_j / 2m$, with $A_{ij}$ and $k_i$ denoting the network adjacency matrix and degree of vertex i, respectively. Partitions of the network into more than two communities can be constructed recursively [40]. Note that modularity-based community partitions can be efficiently computed for large social networks, and can be constructed even with incomplete network topology data [39].

With this definition in hand, we are in a position to present the first candidate predictive feature nominated by the theoretical predictability assessment: the presence of early diffusion activity in numerous distinct network communities should be a reliable predictor that the ultimate reach of the diffusion will be large (see Appendix Two). In what follows, propagation dynamics which possess this characteristic will be said to exhibit *significant early dispersion across network communities.* Note that this measure should be more predictive than the early volume of diffusion activity (the latter has recently become a fairly standard measure [e.g., 19,20]). A cartoon illustrating the basic idea behind this result is given in Figure 2.

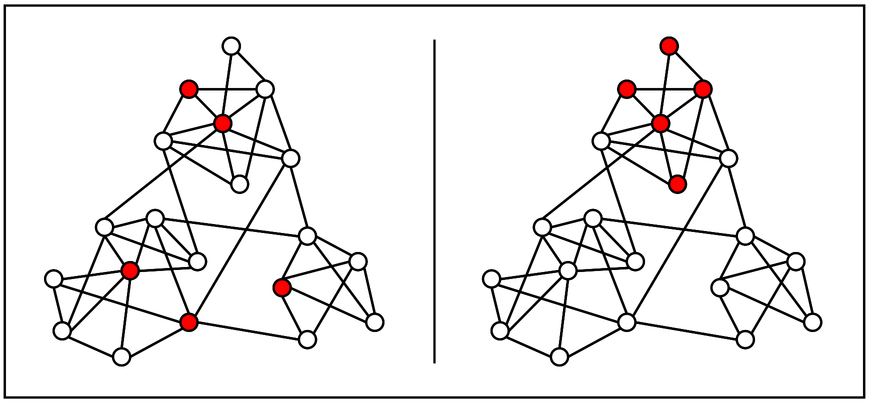


**Figure 2.** Early dispersion across communities is predictive. The cartoon illustrates the predictive feature associated with community structure: social diffusion initiated with five "seed" individuals is much more likely to propagate widely if these seeds are dispersed across three communities (left) rather than concentrated within a single community (right). Note that in Appendix Two this result is established for networks of realistic scale and not simply for "toy" networks like the one shown here.

Analogously to the situation with network communities, there exists a wide range of qualitative and quantitative descriptions of the core-periphery structure found in real-world networks. Here we adopt the characterization of network core-periphery which results from *k-shell decomposition*, a well-established technique in graph theory that is summarized in, for instance, [41]. To partition a network into its k-shells, one first removes all vertices with degree one, repeating this step if necessary until all remaining vertices have degree two or higher; the removed vertices constitute the 1-shell. Continuing in the same way, all vertices with degree two (or less) are recursively removed, creating the 2-shell. This process is repeated until all vertices have been assigned to a k-shell. The shell with the highest index, the $k_{max}$-shell, is deemed to be the core of the network.

Given this definition, we are in a position to report the second candidate predictive feature nominated by our theoretical predictability assessment: early diffusion activity within the network $k_{max}$-shell should be a reliable predictor that the ultimate reach of the diffusion will be significant (see Appendix Two). In particular, this measure should be more predictive than the early volume of diffusion activity. An intuitive illustration of this result is depicted in Figure 3.

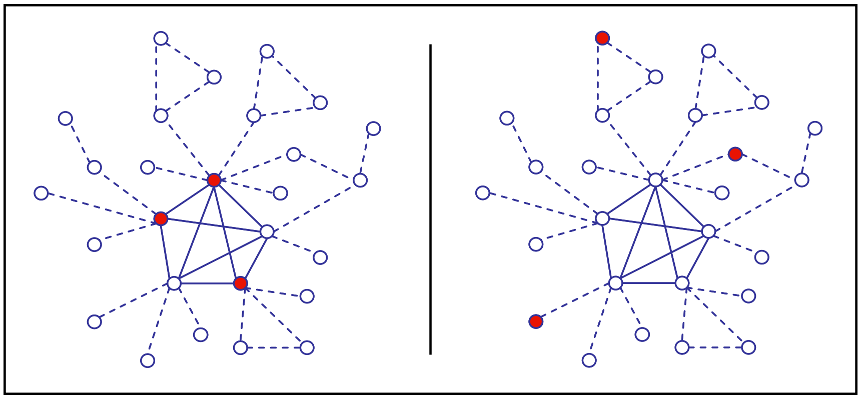


**Figure 3.** Early diffusion within the core is predictive. The cartoon illustrates the predictive feature associated with k-shell structure: social diffusion initiated with three "seed" individuals is much more likely to propagate widely if these seeds reside within the network's core (left) rather than at its periphery (right). Note that in Appendix Two this result is established for networks of realistic scale and not simply for "toy" networks like the one shown here.

### 2.4 Early Warning Method

We are now in a position to present an early warning method which is capable of accurately predicting, very early in the lifecycle of a diffusion process of interest, whether or not the process will propagate widely. We adopt a machine learning-based classification approach to this problem: given a triggering incident, one or more information sources which reflect the reaction to this trigger by a population of interest, and a definition for what constitutes an "alarming" reaction, the goal is to learn classifier that accurately predicts, as early as possible, whether or not reaction to the event will ultimately become alarming. The classifier used in the empirical studies described in this paper is the Avatar ensembles of decision trees (A-EDT) algorithm [42]. Other classification algorithm were also explored to allow the robustness of the proposed early warning approach to be evaluated, and these alternative methods produced qualitatively similar results [39]. Prediction accuracy in all tests is estimated using standard N-fold cross-validation, in which the set of diffusion events of interest is randomly partitioned into N subsets of equal size, and the A-EDT algorithm is successively "trained" on N−1 of the subsets and "tested" on the held-out subset in such a way that each of the N subsets is used as the test set exactly once.

A key aspect of the proposed approach to early warning analysis is determining which characteristics of the social diffusion event of interest, if any, possess exploitable predictive power. We consider three classes of features:

- *intrinsics-based features* – measures of the inherent properties and attributes of the "object" being diffused;
- *simple dynamics-based features* – metrics which capturing simple properties of the diffusion dynamics, such as the early extent of the diffusion and the rate at which the diffusion is propagating;
- *network dynamics-based features* – measures that characterize the way the early diffusion is progressing relative to topological properties of the underlying social and information networks (e.g., community structure).

Consider, as an illustrative example, the diffusion of "memes", that is, short textual phrases which propagate relatively unchanged online (e.g., 'lipstick on a pig'). Suppose it is of interest to predict which memes will "go viral", appearing in thousands of blog posts, and which will not. In this case, intrinsic-based features could include language measures, such as the sentiment or emotion expressed in the text surrounding the memes in blog posts or news articles. Simple dynamics-based features for memes might measure the cumulative number of posts or articles mentioning the meme of interest at some early time $\tau$ and the rate at which this volume is increasing. Network dynamics-based features might count the cumulative number of network communities in a blog graph $G_B$ that contain at least one post which mentions the meme by time $\tau$ and the number of blogs in the $k_{max}$-shell of $G_B$ that, by time $\tau$, contain at least one post mentioning the meme. Alternatively, in the case of an epidemic, the intrinsic-based features could include the infectivity of the pathogen, simple dynamics-based features might capture the number of individuals infected by the disease in the early stages of the outbreak, and network dynamics-based features could include metrics that characterize the way the epidemic is progressing over the communities of relevant social and transportation networks.

The proposed approach to early warning analysis is to collect features from these classes for the event of interest, input the feature values to the (trained) A-EDT classifier, and then run the classifier to generate the warning prediction (i.e., a forecast that the event is expected to become 'alarming' or remain 'not alarming'). In the algorithm presented below this procedure in specified in general terms; more specific instantiations of the procedure are presented in the discussions of the three case studies in Section 3. In what follows it is assumed that the primary source of information concerning the event of interest is social media, as that is emerging as a very useful data source for predictive analysis [e.g., 17-24,26,27]. However, the analytic process is quite similar when other data sources (e.g., intelligence reporting) are employed [24].

Thus we have the following early warning algorithm:

**Algorithm EW**

Given: a triggering incident, a definition for what constitutes an 'alarming' reaction, and a set of social media sites (e.g., blogs) B which are relevant to early warning task.

Initialization: train the A-EDT classifier on a set of events which are qualitatively similar to the triggering event of interest and are labeled as 'alarming' or 'not alarming' according to the definition given above (see the case study discussions for additional details on this training process).

Procedure:

1. Assemble a lexicon of keywords L that pertain to the triggering event under study.
2. Conduct a sequence of blog graph crawls and construct a time series of blog graphs $G_B(t)$. For the lexicon L and each time period t, label each blog in $G_B(t)$ as 'active' if it contains a post mentioning any of the keywords in L and 'inactive' otherwise.
3. Form the union $G_B = \cup_t G_B(t)$, partition $G_B$ into network communities and into k-shells, and map the partition element structure of $G_B$ back to each of the graphs $G_B(t)$.
4. Compute the values of appropriate measures for the intrinsics, simple dynamics, and network dynamics features for each of the graphs $G_B(t)$.
5. Apply the A-EDT classifier to the available time series of features, that is, the features obtained from the sequence of blog graphs $\{G_B(t_0), \ldots, G_B(t_p)\}$, where $t_0$ and $t_p$ are the triggering event time and present time, respectively. Issue an early warning alert if the classifier output is 'alarming'.

We now offer additional details concerning this procedure; more application-specific discussions of the methodology are provided in the case studies in Section 3. Identifying appropriate keywords in Step 1 can be accomplished with the help of subject matter experts and also through various automated means (e.g., via meme analysis [38,27]). Step 2 is by now standard, and various tools exist which can perform these tasks [e.g., 43]. In Step 3, blog network communities are identified with a modularity-based community extraction algorithm applied to the blog graph [40], while the decomposition of the graph into its k-shells is achieved through standard methods [41]. The particular choices of metrics for the intrinsics, simple dynamics, and network dynamics features computed in Step 4 tend to be problem specific, and typical examples are given in the case studies below. It is worth noting, however, that we have found it useful in a range of applications to quantify the dispersion of activity over the communities of $G_B(t)$ using a blog entropy measure BE:

$$BE(t) = -\Sigma_i f_i(t) \log(f_i(t)),$$

where $f_i(t)$ is the fraction of total posts containing one or more keywords and made during interval t which occur in community i. Finally, in Step 5 the feature values obtained in Step 4 serve as inputs to the A-EDT classifier and the output is used to decide whether an alert should be issued.

## 3. Case Studies

This section applies Algorithm EW to three early warning case studies involving social phenomena that have proved to be both practically important and challenging to analyze: 1.) diffusion of information through social media, 2.) mobilization/protest events response to "triggering" incidents, and 3.) planning/coordination/execution of politically-motivated cyber attacks.

### 3.1 Case Study One: Meme Diffusion

The goal of this case study is to apply Algorithm EW to the task of predicting whether or not a given "meme", that is, a short textual phrase which propagates relatively unchanged online, will "go viral". Our main source of data on meme dynamics is the publicly available datasets archived at http://memetracker.org [44] by the authors of [38]. Briefly, the archive [44] contains time series data characterizing the diffusion of ~70 000 memes through social media and other online sites during the five month period between 1 August and 31 December 2008. We are interested in using Algorithm EW to distinguish successful and unsuccessful memes early in their lifecycle. More precisely, the task of interest is to classify memes into two groups – those which will ultimately be successful (acquire more than S posts) and those that will be unsuccessful (attract fewer than U posts) – very early in the meme lifecycle.

To support an empirical evaluation of the utility of Algorithm EW for this problems, we downloaded from [44] the time series data for slightly more than 70 000 memes. These data contain, for each meme M, a sequence of pairs $(t_1, URL_1)_M$, $(t_2, URL_2)_M$, …, $(t_T, URL_T)_M$, where $t_k$ is the time of appearance of the kth blog post or news article that contains at least one mention of meme M, $URL_k$ is the URL of the blog or news site on which that post/article was published, and T is the total number of posts that mention meme M. From this set of time series we randomly selected 100 "successful" meme trajectories, defined as those corresponding to memes which attracted at least 1000 posts during their lifetimes, and 100 "unsuccessful" meme trajectories, defined as those whose memes acquired no more than 100 total posts. It is worth noting that, in assembling the data in [44], all memes which received fewer than 15 total posts were deleted, and that ~50% of the remaining memes have <50 posts; thus the large majority of memes are unsuccessful by our definition (as well as according to the criteria of most applications [38,27]).

Two other forms of data were collected for this study: 1.) a large Web graph which includes websites (URLs) that appear in the meme time series, and 2.) samples of the text surrounding the memes in the posts which contain them. More specifically, we sampled the URLs appearing in the time series for our set of 200 successful and unsuccessful memes and performed a Web crawl that employed these URLs as "seeds". This procedure generated a Web graph, denoted $G_B$, that consists of approximately 550 000 vertices/websites and 1.4 million edges/hyperlinks, and includes essentially all of the websites which appear in the meme time series. To obtain samples of text surrounding memes in posts, we randomly selected ten posts for each meme and then extracted from each post the paragraph which contains the first mention of the meme.

Recall that Algorithm EW employs three types of features: intrinsics-based, simple dynamics-based, and network dynamics-based. We now describe the instantiation of each of these feature classes for the meme problem. Consider first the intrinsics-based features, which for the meme application become language-based measures. Each "document" of text surrounding a meme in its (sample) posts is represented by a simple "bag of words" feature vector $x \in \Re^{|V|}$, where the entries of x are the frequencies with which the words in the vocabulary set V appear in the document. A very simple way to quantify the sentiment or emotion of a document is through the use of appropriate lexicons. Let $s \in \Re^{|V|}$ denote a lexicon vector, in which each entry of s is a numerical "score" quantifying the sentiment/emotion intensity of the corresponding word in the vocabulary V. The aggregate sentiment/emotion score of document x can be computed as

$$\text{score}(x) = s^T x \,/\, s^T 1,$$

where 1 is a vector of ones. Thus score(.) estimates the sentiment or emotion of a document as a weighted average of the sentiment or emotion scores for the words comprising the document. (Note that if no sentiment or emotion information is available for a particular word in V then the corresponding entry of s is set to zero.)

To characterize the emotion content of a document we use the Affective Norms for English Words (ANEW) lexicon, which consists of 1034 words that were assigned numerical scores with respect to three emotional "axes" – happiness, arousal, and dominance – by human subjects [45]. Previous work had identified this set of words to bear meaningful emotional content [45]. Positive or negative sentiment is quantified by employing the "IBM lexicon", a collection of 2968 words that were assigned {positive, negative} sentiment labels by human subjects [46]. This simple approach generates four language features for each meme: the happiness, arousal, dominance, and positive/negative sentiment of the text surrounding that meme in the (sample) posts containing it. As a preliminary test, we computed the mean emotion and sentiment of content surrounding the 100 successful and 100 unsuccessful memes in our dataset. On average the text surrounding successful memes is happier, more active, more dominant, and more positive than that surrounding unsuccessful memes, and this difference is statistically significant ($p<0.0001$). Thus it is at least plausible that these four language features may possess some predictive power regarding meme success.

Consider next two simple dynamics-based features, defined to capture the basic characteristics of the early evolution of meme post volume:

- #posts($\tau$) – the cumulative number of posts mentioning the given meme by time $\tau$ (where $\tau$ is small relative to the typical lifespan of memes);
- post rate($\tau$) – a simple estimate of the rate of accumulation of such posts at time $\tau$.

Here we adopt a simple finite difference definition for post rate given by post rate($\tau$) = (#posts($\tau$) – #posts($\tau$/2)) / ($\tau$/2); of course, more robust rate estimates could be used.

The simple dynamics-based measures of early meme diffusion defined above, while potentially useful, do not characterize the manner in which a meme propagates over the underlying social or information networks. Recall that the predictability assessment summarized in Section 2.3 suggests that both early dispersion of diffusion activity across network communities and early diffusion activity within the network core ought to be predictive of meme success. The insights offered by this theoretical analysis motivate the definition of two network dynamics-based features for meme prediction:

- community dispersion($\tau$) – the cumulative number of network communities in the blog graph $G_B$ that, by time $\tau$, contain at least one post which mentions the meme;
- #k-core blogs($\tau$) – the cumulative number of blogs in the $k_{max}$-shell of blog graph $G_B$ that, by time $\tau$, contain at least one post which mentions the meme.

These quantities can be efficiently computed using fast algorithms for partitioning a graph into its communities and for identifying a graph's $k_{max}$-shell [39]. Thus these features are readily computable even for very large graphs.

We now summarize the results of this case study. First, using only the four language features with the A-EDT classifier to predict which memes will be successful yields a prediction accuracy of 66.5% (ten-fold cross-validation). Since simply guessing "successful" for all memes gives an accuracy of 50%, it can be seen that these simple language intrinsics are not very predictive. For completeness it is mentioned that the ANEW score for "arousal" and the IBM measure of sentiment are the most predictive of these four features. In contrast, the features characterizing the early network dynamics of memes possess significant predictive power, and in fact are useful even if only very limited early time series is available for use in prediction. More quantitatively, applying Algorithm EW with the four meme dynamics features produces the following results (ten-fold cross-validation):

- $\tau = 12$hr, accuracy = 84%, most predictive features: 1.) community dispersion, 2.) #k-core blogs, 3.) #posts;
- $\tau = 24$hr, accuracy = 92%, most predictive features: 1.) community dispersion, 2.) post rate, 3.) #posts;
- $\tau = 48$hr, accuracy = 94%, most predictive features: 1.) community dispersion, 2.) post rate, 3.) #posts.

These results show that useful predictions can be obtained *within the first twelve hours* after a meme is detected (this corresponds to 0.5% of the average meme lifespan), and that accurate prediction is possible after about a day or two. Note also that, as has been found with other social dynamics phenomena [e.g., 16-18], dynamics features appear to be more predictive than "intrinsics", at least for the features employed here.

It is worth mentioning that the fact that a particular meme goes viral does not imply that it will influence behavior in the real world. The next two case studies focus on the important issue of behavioral consequences of information diffusion.

### 3.2 Case Study Two: Mobilization and Protest

There is considerable interest to develop methods for distinguishing successful mobilization and protest events, that is, mobilizations that become large and self-sustaining, from unsuccessful ones early in their lifecycle. It is natural to pose this question as an early warning problem and to approach it using Algorithm EW. In order to examine the efficacy of this approach, we collected together fourteen recent events, each of which appeared at the outset to have the potential to trigger significant protests. This set of events contains seven triggering incidents which ultimately led to substantial mobilization, including massive protests and significant violence, and seven triggers with reactions that subsided quickly with essentially no violence. Taken together, these events provide a useful setting for testing the applicability of Algorithm EW to mobilization/protest phenomena.

The events employed in this study are listed below.

Triggers leading to significant mobilization/protest:

- Quran desecration, May 2005;
- first Danish cartoons, September 2005 to March 2006;
- Egypt DVD release, October 2005;
- France riots, October and November 2005;
- anti-Ahmadiyya protests, June and July 2008;
- U.S Republican National Convention, September 2008;
- Israel/Gaza event, December 2008 to January 2009.

Triggers not leading to significant mobilization/protest:

- Abu Ghraib news release, April and May 2004;
- Pope lecture, September 2006;
- Salman Rushdie knighting, June 2007;
- second Danish cartoons, February 2008;
- U.S. Democratic National Convention, August 2008,
- Bali bombers execution, November 2008;
- Jakarta bombings/NM Top blog post, July 2009.

This list is intended merely to identify the fourteen events under study; additional information concerning each incident is given in [39] and the references therein.

As a preliminary examination of the possibility to obtain useful early warning indicators from analysis of social media discussions of these events, we performed Steps 1-4 of Algorithm EW and then plotted the time series for two quantities: 1.) the volume of blog posts mentioning keywords relevant to the events (these keywords were obtained through a simple news search [39]), and 2.) the blog entropy measure $BE(t) = -\Sigma_i f_i(t) \log(f_i(t))$ associated with the way online mentions of the keywords diffused over the blog graph. Illustrative time series plots are shown in Figure 4. Observe that in the case of the first Danish cartoons event (plot at right) the BE of relevant discussions (blue curve) experiences a dramatic increase a few weeks before the corresponding increase in volume of blog discussions (red curve); this latter increase, in turn, takes place before any violence. In contrast, in the case of the pope event (plot at left), BE of blog discussions is small relative to the cartoons event, and any increase in this measure lags discussion volume. Similar time series plots are obtained for the other twelve events, suggesting that network dynamics-based features, such as dispersion of discussions across blog network communities, may be a useful early indicator for large mobilization events.

To examine this possibility more carefully, we applied Algorithm EW to the task of distinguishing triggers which led to large protests from those that did not. For simplicity, in this case study we did not use any intrinsics-based features (e.g., language metrics) in the A-EDT classifier, and instead relied upon the four dynamics-based features defined in Case Study One. In the case of the seven triggering events which led to protest behavior, the blog data made available to Algorithm EW was limited to posts made during the eight week period which ended two weeks before the protests began. For the seven triggers which did not lead to protests, the blog data included all posts collected during the eight week period immediately following the triggering event.

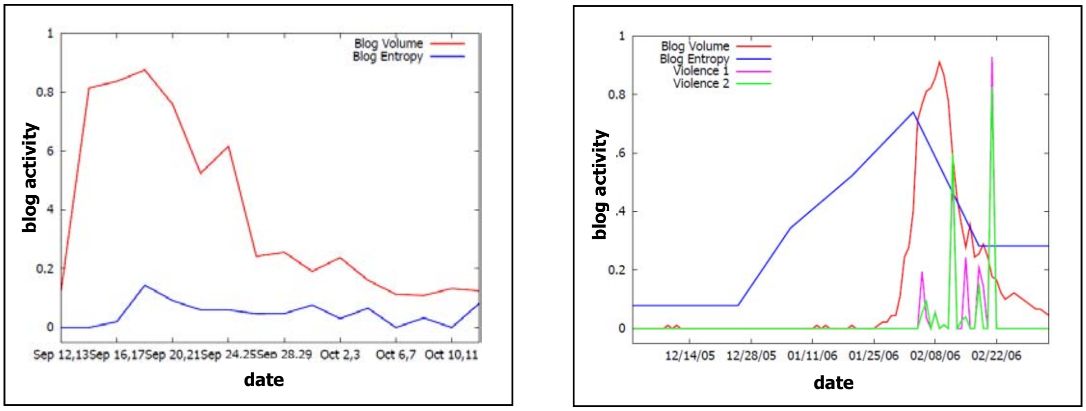


**Figure 4.** Sample results for mobilization/protest case study. The illustrative time series plots shown correspond to the pope event (left) and first Danish cartoons event (right). In each plot, the red curve is blog volume and the blue curve is blog entropy; the Danish cartoon plot also shows two measures of violence (cyan and magenta curves). Note that while the volume and violence data are scaled to allow multiple data sets to be graphed on each plot, the scale for entropy is consistent across plots to enable cross-event comparison.

Because the set of events in this case study included only fourteen incidents, we applied Algorithm EW with two-fold cross-validation. More specifically, the set of incidents was randomly partitioned into two equal subsets, the algorithm was trained on one subset of seven incidents and tested on the other subset, and then the roles of the two data sets were switched. In this evaluation Algorithm EW achieved *perfect* accuracy, correctly distinguishing the 'protest' and 'non-protest' triggers. An examination of the predictive power of the four features used as inputs to the A-EDT classifier reveals that, as suggested by Figure 4, the community dispersion feature was the most predictive measure.

### 3.2 Case Study Three: Cyber Attack Early Warning

This case study explores the ability of Algorithm EW to provide reliable early warning for politically-motivated distributed denial-of-service (DDoS) attacks. Toward this end, we first identified a set of Internet "disturbances" that included examples from three distinct classes of events:

1. successful politically-motivated DDoS attacks – these are the events for which Algorithm EW is intended to give warning with sufficient lead time to allow mitigating actions to be taken;
2. natural events which disrupt Internet service – these are disturbances, such as earthquakes and electric power outages, that impact the Internet but for which it is known that no early warning signal exists in social media;
3. quiet periods – these are periods during which there is social media "chatter" concerning impending DDoS attacks but ultimately no (successful) attacks occurred.

Including in the case study events selected from these three classes is intended to afford a fairly comprehensive test of Algorithm EW. For instance, these classes correspond to 1.) the domain of interest (DDoS attacks), 2.) a set of disruptions which impact the Internet but have no social media warning signal, and 3.) a set of "non-events" which do not impact the Internet but do possess putative social media warning signals (online discussion of DDoS attacks).

We selected twenty events from these three classes:

Politically-motivated DDoS attacks:

- Estonia event in April 2007;
- CNN/China incident in April 2008;
- Israel/Palestine conflict event in January 2009;
- DDoS associated with Iranian elections in June 2009;
- WikiLeaks event in November 2010;
- Anonymous v. PayPal, etc. attack in December 2010;
- Anonymous v. HBGary attack in February 2011.

Natural disturbances:

- European power outage in November 2006;
- Taiwan earthquake in December 2006;
- Hurricane Ike in September 2008;
- Mediterranean cable cut in January 2009;
- Taiwan earthquake in March 2010;
- Japan earthquake in March 2011.

Quiet periods:

Seven periods, from March 2005 through March 2011, during which there were discussions in social media of DDoS attacks on various U.S. government agencies but no (successful) attacks occurred.

For brevity a detailed discussion of these twenty events is not given here; the interested reader is referred to [39] and the references therein for additional information on these disruptions.

We collected two forms of data for each of the twenty events: *cyber data* and *social data*. The cyber data consist of time series of routing updates which were issued by Internet routers during a one month period surrounding each event. More precisely, these data are the Border Gateway Protocol (BGP) routing updates exchanged between gateway hosts in the Autonomous System network of the Internet. The data was downloaded from the publicly-accessible RIPE collection site [47] using the process described in [48] (see [48] for additional details and background information on BGP routing dynamics). The temporal evolution of the volume of BGP routing updates (e.g., withdrawal messages) gives a coarse-grained measure of the timing and magnitude of large Internet disruptions and thus offers a simple and objective way to characterize the impact of each of the events in our collection. The social data consist of time series of social media mentions of cyber attack-related keywords and memes detected during a one month period surrounding each of the twenty events. These data were collected using the procedure specified in Algorithm EW.

As in the preceding case study, we performed a preliminary examination of the possibility to obtain useful early warning indicators from analysis of social media discussions by completing Steps 1-4 of Algorithm EW and plotting the time series for two quantities: 1.) the volume of blog posts mentioning keywords relevant to the events (these keywords were obtained through a simple news search [39]), and 2.) the blog entropy measure $BE(t) = -\Sigma_i f_i(t) \log(f_i(t))$ associated with the way online mentions of the keywords diffused over the blog graph. Illustrative time series plots corresponding to two events in the case study, the WikiLeaks DDoS attack in November 2010 and Japan earthquake in March 2011, are shown in Figure 5. Observe that the time series of BGP routing updates are similar for the two events, with each experiencing a large "spike" at the time of the event. The time series of blog post volume are also similar across the two events, with each showing modest volume prior to the event and displaying a large spike in activity at event time. However, the time series for blog entropy are quite distinct for the two events. Specifically, in the case of the WikiLeaks DDoS the blog entropy (blue curve in Figure 5) experiences a dramatic increase several days before the event, while in the case of the Japan earthquake blog entropy is small for the entire collection period. Similar social media behavior is observed for all events in the case study, suggesting that network dynamics-based features, such as dispersion of discussions across blog network communities, may be a useful early indicator for large mobilization events.

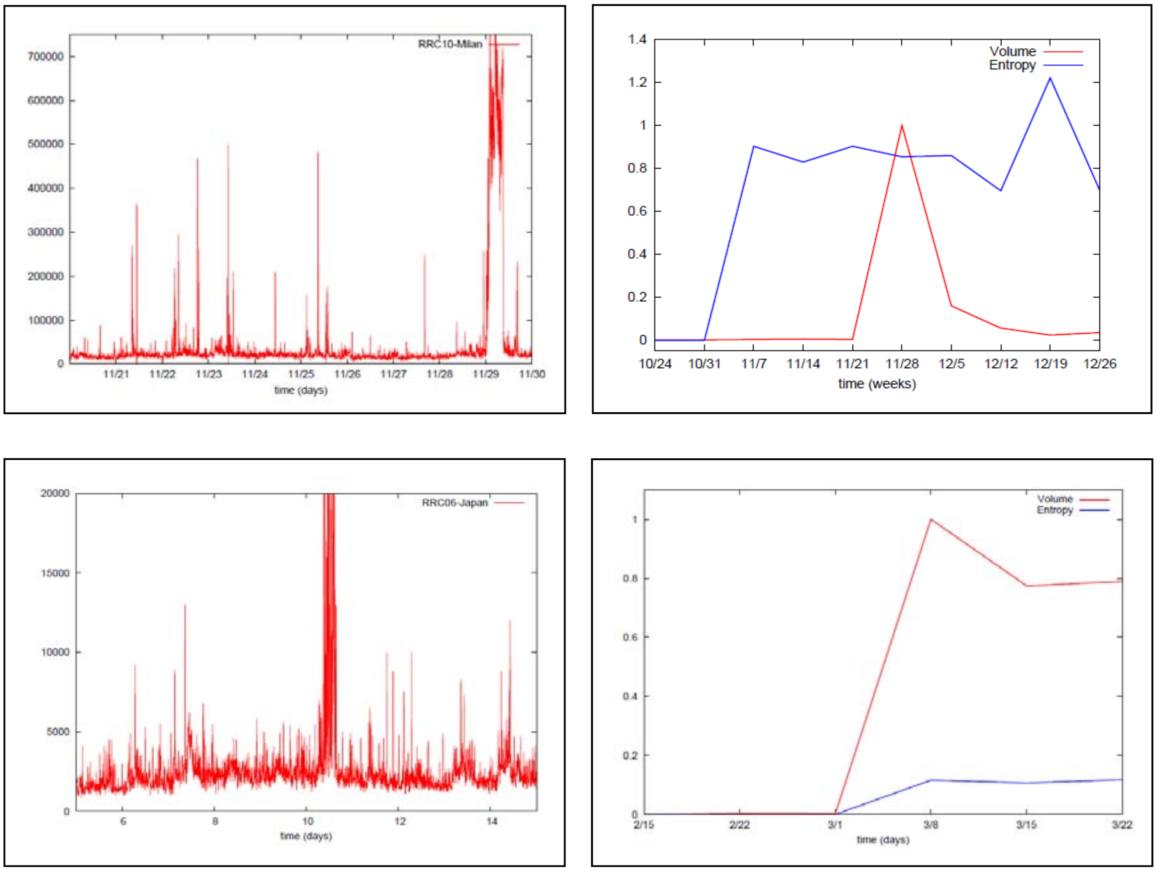


**Figure 5.** Sample results for the DDoS early warning case study. The illustrative time series plots shown correspond to the WikiLeaks event in November 2010 (top row) and the Japan earthquake in March 2011 (bottom row). For each event, the plot at left is the time series of BGP routing updates (note the large increase in updates triggered by the event). The plot at the right of each row is the time series of the social media data, with the red curve showing blog post volume and the blue curve depicting blog entropy. Note that while post volume is scaled for convenient visualization, the scale for entropy is consistent across plots to allow cross-event comparison.

To examine this possibility more carefully, we applied Algorithm EW to the task of distinguishing the seven DDoS attacks from the thirteen other events in the set. For simplicity, in this case study we did not use any intrinsics-based features (e.g., language metrics) in the A-EDT classifier, and instead relied upon the four dynamics-based features defined in Case Study One. Because the set of events in this case study included only twenty incidents, we applied Algorithm EW with two-fold cross-validation, exactly as described in Case Study Two. In the case of DDoS events, the blog data made available to Algorithm EW was limited to posts made during the five week period which ended one week before the attack. For the six natural disturbances, the blog data included all posts collected during the six week period immediately prior to the event, while in the case of the seven non-events, the blog data included the posts collected during a six week interval which spanned discussions of DDoS attacks on U.S. government agencies.

In this evaluation, Algorithm EW achieved *perfect* accuracy, correctly distinguishing the 'attack' and 'non-attack' events. If the test is made more difficult, so that the blog data made available to Algorithm EW for attack events is limited to a four week period that ends two weeks before the attack, the proposed approach still achieves 95% accuracy, An examination of the predictive power of the four features used as inputs to the A-EDT classifier reveals that, as suggested by Figure 5, the community dispersion feature was the most predictive measure. It is worth emphasizing that, in this case study, accurately distinguishing 'attack' from 'non-attack' events is equivalent to providing practically-useful early warning for attack events, because the data which serves as input to Algorithm EW reflects online discussions that took place *prior to* the events under investigation.

## 4. Conclusions

This paper presents a new approach to early warning analysis for social diffusion events. We begin by introducing a biologically-inspired S-HDS model for social dynamics on multi-scale networks, and then perform stochastic reachability analysis with this model to show that the outcomes of social diffusion processes may depend crucially upon the way the early dynamics of the process interacts with the underlying network's meso-scale topological structures. This theoretical finding provides the foundations for developing a machine learning algorithm that enables accurate early warning analysis for diffusion events. The utility of the warning algorithm, and the power of network-based predictive metrics, are demonstrated through empirical case studies involving meme propagation, large-scale protests events, and politically-motivated cyber attacks.

## 5. Acknowledgements

This research was supported by the U.S. Department of Defense, the U.S. Department of Homeland Security, The Boeing Company, and the Laboratory Directed Research and Development program at Sandia National Laboratories. Fruitful discussions regarding aspects of this work with Curtis Johnson of Sandia National Laboratories, Paul Ormerod of Volterra Partners, and Anne Kao of Boeing are gratefully acknowledged.

## 6. References

[1] Anderson, R. and R. May, *Infectious Diseases of Humans*, Oxford University Press, 1992.

[2] Rogers, E., *Diffusion of Innovations*, Fifth Ed., Free Press, NY, 2003.

[3] Della Porta, D. and M. Diani, *Social Movement*, Second Ed., Blackwell, Oxford, UK, 2006.

[4] Easley, D. and J. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.

[5] Moghadam, A., The Globalization of Martyrdom: Al Qaeda, Salafi Jihad, and the Diffusion of Suicide Attacks, Johns Hopkins University Press, Baltimore, 2008.

[6] Myers, D. and P. Oliver, "The opposing forces diffusion model: The initiation and repression of collective violence", *Dynamics of Asymmetric Conflict*, Vol. 1, pp. 164-188, 2008.

[7] Ackerman, G., et al., "Anticipating rare events: Can acts of terror, use of weapons of mass destruction, or other high profile acts be anticipated?", DoD White Paper, November 2008.

[8] Krueger, A. and J. Maleckova, "Attitudes and action: Public opinion and the occurrence of international terrorism", *Science*, Vol. 325, pp. 1535-1536, 2009.

[9] Bergin, A., S. Osman, C. Ungerer, and N. Yasin, "Countering internet radicalization in Southeast Asia", ASPI Special Report, March 2009.

[10] Chen, H., C. Yang, M. Chau, and S. Li (Editors), *Intelligence and Security Informatics*, Lecture Notes in Computer Science, Springer, Berlin, 2009.

[11] Proc. 2010 IEEE International Conference on Intelligence and Security Informatics, Vancouver, BC, Canada, May 2010.

[12] O'Brien, S., "Crisis early warning and decision support: Contemporary approaches and thoughts on future research", *International Studies Review*, Vol. 12, pp. 87-104, 2010.

[13] Ward, M., B. Greenhill, and K. Bakke, "The perils of policy by p-value: Predicting civil conflict", *J. Peace Research*, Vol. 47, pp. 363-375, 2010.

[14] Walls, W., "Modeling movie success when 'nobody knows anything': Conditional stable-distribution analysis of film returns", *J. Cultural Economics*, Vol. 29, pp. 177-190, 2005.

[15] Salganik, M., P. Dodds, and D. Watts, "Experimental study of inequality and unpredictability in an artificial cultural market", *Science*, Vol. 311, pp. 854-856, 2006.

[16] Colbaugh, R. and K. Glass, "Predictability and prediction of social processes", *Proc. 4th Lake Arrowhead Conference on Human Complex Systems*, Arrowhead, CA, April 2007.

[17] Colbaugh, R. and K. Glass, "Predictive analysis for social processes I: Multi-scale hybrid system modeling, and II: Predictability and warning analysis", *Proc. 2009 IEEE Multi-Conference on Systems and Control*, Saint Petersburg, Russia, July 2009.

[18] Colbaugh, R., K. Glass, and P. Ormerod, "Predictability of 'unpredictable' cultural markets", *Proc. 105th Annual Meeting of the American Sociological Association*, Atlanta, GA, August 2010.

[19] Asur, S. and B. Huberman, "Predicting the future with social media", *Proc. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Toronto, Ontario, Canada, September 2010.

[20] Goel, S., J. Hofman, S. Lahaie, D. Pennock, and D. Watts, "Predicting consumer behavior with Web search", *Proc. National Academy of Sciences USA*, Vol. 107, pp. 17486-17490, 2010.

[21] Bollen, J., H. Mao, and X. Zeng, "Twitter mood predicts the stock market", arXiv preprint, October 2010.

[22] Tumasjan, A., T. Sprenger, P. Sandner, and I. Welpe, "Predicting elections with Twitter: What 140 characters reveal about political sentiment", *Proc. 4th International AAAI Conference on Weblogs and Social Media*, Washington, DC, May, 2010.

[23] Colbaugh, R. and K. Glass, "Early warning analysis for social diffusion events", *Proc. 2010 IEEE International Conference on Intelligence and Security Informatics*, Vancouver, BC Canada, May 2010.

[24] Colbaugh, R., K. Glass, and J. Gosler, "Some intelligence analysis problems and their graph formulations", *J. Intelligence Community Research and Development*, Paper 315, Permanently available on Intelink, June 2010.

[25] Christakis, N. and J. Fowler, "Social network sensors for early detection of contagious outbreaks", *PLoS ONE*, Vol. 5, e12948, 2010.

[26] Lerman, K. and T. Hogg, "Using stochastic models to describe and predict social dynamics of Web users", arXiv preprint, October 2010.

[27] Colbaugh, R. and K. Glass, "Detecting emerging topics and trends via predictive analysis of 'meme' dynamics", *Proc. 2011 AAAI Spring Symposium Series*, Palo Alto, CA, March 2011.

[28] Colbaugh, R. and K. Glass, "Proactive defense for evolving cyber threats", *Proc. 2011 IEEE International Conference on Intelligence and Security Informatics*, Beijing, China, July 2011.

[29] Uhrmacher, A., D. Degering, and B. Zeigler, "Discrete event multi-level models for systems biology", in *Trans. Computational Systems Biology*, LNBI 3380, Springer, 2005.

[30] El-Samad, H., S. Prajna, A. Papachristodoulou, J. Doyle, and M. Khammash, "Advanced methods and algorithms for biological networks analysis", *Proc. IEEE*, Vol. 94, pp. 832-853, 2006.

[31] Julius, A., A. Halasz, M. Sakar, H. Rubin, V. Kumar, and G. Pappas, "Stochastic modeling and control of biological systems: The lactose regulation system of *Escherichia coli* ", *IEEE Trans. Automatic Control*, Vol. 53, pp. 51-65, 2008.

[32] Lygeros, J., et al., "Stochastic hybrid modeling of DNA replication across a complete genome", *Proc. National Academy of Sciences USA*, Vol. 105, pp. 12295-12300, 2008.

[33] Yuan, C., X. Mao, and J. Lygeros, "Stochastic hybrid delay population dynamics: Well-posed models and extinction", *J. Biological Dynamics*, Vol. 3, pp. 1-21, 2009.

[34] Bujorianu, M., J. Lygeros, and M. Bujorianu, "Toward a general theory of stochastic hybrid systems", e-print, University of Twente, The Netherlands, March 2008.

[35] Doyle, J. and M. Csete, "Architecture, constraints, and behavior", *Proc. National Academy of Sciences USA*, in press.

[36] Newman, M., "The structure and function of complex networks", *SIAM Review*, Vol. 45, pp. 167-256, 2003.

[37] Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Second Edition, Springer, New York, 2009.

[38] Leskovec, J., L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle", *Proc. 15th ACM International Conference on Knowledge Discovery and Data Mining*, Paris, France, June 2009.

[39] Colbaugh, R. and K. Glass, "Prediction of social dynamics via social media analytics", Sandia National Laboratories SAND Report, January 2011.

[40] Newman, M., "Modularity and community structure in networks", *Proc. National Academy of Sciences USA*, Vol. 103, pp. 8577-8582, 2006.

[41] Carmi, S., S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir, "A model of Internet topology using the k-shell decomposition", *Proc. National Academy of Sciences USA,* Vol. 104, pp. 11150-11154, 2007.

[42] http://www.sandia.gov/avatar/, accessed July 2010.

[43] Glass, K. and R. Colbaugh, "Web analytics for security informatics", *Proc. European Intelligence and Security Informatics Conference*, Athens, Greece, September 2011.

[44] http://memetracker.org, accessed January 2010.

[45] Bradley, M. and P. Lang, "Affective norms for English words (ANEW): Stimuli, instruction manual, and affective ratings", Technical Report C1, University of Florida, 1999.

[46] Ramakrishnan, G., A. Jadhav, A. Joshi, S. Chakrabarti, and P. Bhattacharyya, "Question answering via Bayesian inference on lexical relations", *Proc. Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, July 2003.

[47] http://data.ris.ripe.net/, last accessed July 2011.

[48] Glass, K., R. Colbaugh, and M. Planck, “Automatically identifying the sources of large Internet events”, *Proc. IEEE International Conference on Intelligence and Security Informatics*, Vancouver, Canada, May 2010.

[49] Hedstrom, P., “Explaining the growth patterns of social movements”, *Understanding Choice, Explaining Behavior*, Oslo University Press, 2006.

[50] Hedstrom, P., R. Sandell, and C. Stern, “Mesolevel networks and the diffusion of social movements: The case of the Swedish Social Democratic Party”, *American J. Sociology*, Vol. 106, pp. 145 -172, 2000.

[51] Bettencourt, L., A. Cintron-Arias, D. Kaiser, and C. Castillo-Chavez, “The power of a good idea: Quantitative modeling of the spread of ideas from epidemiological models”, *Physica A*, Vol. 364, pp. 513-536, 2006.

[52] Candia, J. and K. Mazzitello, “Mass media influence spreading in social networks with community structure”, *J. Statistical Mechanics*, Vol. 7, P07007, 2008.

[53] Kushner, H.*, Stochastic Stability and Control*, Academic Press, NY, 1967.

[54] Papachristodoulou, A., Personal communication, November 2008.

[55] Parrilo, P., Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization, PhD dissertation, California Institute of Technology, 2000.

[56] http://www.cds.caltech.edu/sostools/, accessed July 2007.

## A1. Appendix One: S-HDS Social Diffusion Model

In this Appendix we propose a multi-scale structure for modeling social network dynamics, establish a few facts concerning this representation, and introduce an S-HDS formulation of the model that is well-suited for predictive analysis.

### A1.1 Multi-Scale Social Dynamics Model

In many social situations, people are influenced by the behavior of others, for instance because they seek to obtain the benefits of coordinated actions, infer otherwise inaccessible information, or manage complexity in decision-making. Processes in which observing a certain behavior increases an individual's probability of adopting that behavior are often referred to as *positive externality processes* (PEP), and we use that term here. PEP have been widely studied in the social and behavioral sciences and, more recently, by the informatics and physical sciences communities [e.g., 4]. In particular, social scientists have constructed theories which qualitatively and quantitatively explain these processes and their dependence on social networks [e.g., 2-4, 6, 18, 36, 49-52]. One result of this research is a recognition that the process by which preferences and opinions of individuals become the collective outcome for a group can be complex and subtle, and thus challenging to model and predict. People arrive at their decisions by reacting individually to an environment consisting largely of others who are reacting likewise, and one consequence of this feedback dynamics is that the collective outcome can be quite different from one implied by a simple aggregation of individual preferences.

We model PEP in a manner which explicitly separates the individual, or "micro", dynamics from the collective dynamics. More specifically, we adopt a modeling framework consisting of three modeling scales:

- a *micro-scale*, for modeling the behavior of individuals;
- a *meso-scale*, which represents the interaction dynamics of individuals within the same network partition element (community or core/periphery);
- a *macro-scale*, which characterizes the interaction between partition elements.

We now derive a few properties of the multi-scale model. The micro-scale quantifies the way individuals combine their own inherent preferences regarding the available options with their observations of the behaviors of others to arrive at their chosen courses of action. Interestingly, the dependence of this decision-making process on the social network admits a straightforward characterization. Consider the common and important binary choice setting, in which N agents choose from a set $O = \{0,1\}$ of options based in part on the choices made by others. Let $o_i \in \{0,1\}$ denote the selection of agent i and $o = [o_1 \ldots o_N]^T \in O^N$ represent the vector of choices made by the group. It is reasonable to suppose that agent i chooses between the options probabilistically according to some map $PO_i$: $A_i \times O^N \rightarrow [0,1]$, where $PO_i$ is the probability that agent i chooses option 1, $A_i$ measures i's inherent preference for option 1, and $PO_i$ is nondecreasing in $A_i$. In positive externality situations $PO_i$ should also be "nondecreasing in o" in some sense, and we now make this notion precise. (For notational simplicity in what follows we suppress the dependence of $PO_i$ on $A_i$.)

Because it is defined in such general terms it may appear that the map $PO_i$ could be a very complicated function of the choices of the other agents. In fact, Theorem 1 indicates that this map must be tractable.

**Theorem 1:** Given any $PO_i$ there exists a vector $w_i = [w_{i1} \ldots w_{iN}]^T \in \Re^N$, with $w_{ij} \geq 0$ and $\Sigma_j w_{ij} = b_i$, and a scalar function $r_i$: $[0, b_i] \rightarrow [0,1]$ such that $PO_i(o) = r_i(o^T w_i)$.

**Proof:** It is enough to prove that the $w_{ij}$ can be chosen so $o^Tw_i$: $O^N \rightarrow [0, b_i]$ is injective, since then $r_i$ can be constructed to recover any $PO_i$. One such choice for $w_i$ is $w_i = [2^0\ 2^1 \ldots 2^{N-1}]^T$, as then $o^Tw_i$ provides a unique (binary number) representation for each o. ■

We call $r_i$ the *agent decision function* and $s_i = o^Tw_i$ agent i's *social signal*, and interpret the $w_{ij}$ as defining a weighted social network for the group of N agents. Observe that Theorem 1 quantifies the way social influence is transmitted to an agent by her neighbors and highlights the importance of this signal in the decision-making process. The result also allows a simple characterization of positive externality agent behavior: for such behavior, $r_i$ is nondecreasing in $s_i$.

The micro-scale model structure allows PEP behaviors which appear to be distinct to be represented within a unified setting. For example, the basic model readily accommodates two of the most common sources of PEP: 1.) *utility-oriented externalities*, in which the utility or value of an option is a direct function of the number of others choosing it, and 2.) *information externalities*, which arise from inferences made by an individual about decision-relevant information possessed by others.

**Example A1.1: utility-oriented externalities.** Suppose each agent i has a utility function $u_i$: $O \times [0, b_i] \rightarrow \Re^+$ which depends explicitly on i's social signal $s_i$. The standard, albeit dated, example here is the fax machine, with the utility of owning a fax machine increasing with the number of others who own one. The key quantity considered by agent i when selecting between options 0 and 1 is the utility difference between the options, $\Delta u_i(s_i) = u_i(1,s_i) - u_i(0,s_i)$. In positive externality situations $\Delta u_i$ is increasing in $s_i$, and there exists a *threshold* social signal value s*, possibly with $s^* < 0$ or $s^* > b_i$, such that a utility maximizing agent will choose option 0 if $s_i < s^*$ and option 1 if $s_i \geq s^*$.

**Example A1.2: information externalities.** Suppose the utility to agent i of each option is independent of the number of other agents choosing that option but there exists uncertainty regarding this utility. To be concrete, assume that agent i's utility depends on the "state of world" $w \in \{w_0, w_1\}$, so that $u_i = u_i(o_i,w)$, and there exists uncertainty regarding w. In this case, agent i may observe others' decisions in order to infer w and then choose the option which maximizes his utility for this world state (as when a tourist chooses a crowded restaurant over an empty one in an unfamiliar city). Consider, for instance, the decision of whether to adopt an innovation of uncertain quality, and let the world state $w_1$ signify that innovation quality is such that adopting maximizes utility. In this situation it is reasonable for agent i to maximize *expected* utility and choose the option (adopt or not) $o_i^* = \text{argmax}_{o \in O} \Sigma_{w \in W} P(w \mid s_i)\ u_i(o_i,w)$. If agent i uses Bayesian inference to estimate $P(w_1 \mid s_i)$ then we have a positive externality decision process and there exists a threshold value s* for the social signal such that agent i will choose option 0 if $s_i < s^*$ and option 1 if $s_i \geq s^*$ [17].

It can be seen that in these examples, different positive externality "drivers" lead to equivalent (threshold) micro-scale models.

Taken together, the meso- and macro-scale components of the proposed modeling framework quantify the way agent decision functions interact to produce collective behavior at the population level. For convenience of exposition, in this Appendix we focus on network communities as the meso-scale structure of interest; however, all of the modeling results derived here also hold for the case of core-periphery structure . The role of the meso-scale model is to quantify and illuminate the manner in which agent decision functions interact *within* social network communities, while the macro-scale model characterizes the interactions of agents *between* communities. The primary assumption is that interactions between individuals within social network communities can be modeled as "fully-mixed" – all pairwise interactions between individuals within a network community are equally likely – while interactions between communities are constrained by the network defining the relationships between the communities. We argue below that this assumption is reasonable and useful.

One advantage of identifying a scale at which agent interaction is (approximately) homogeneous is that this enables the leveraging of an extensive literature on collective dynamics. To be concrete, we derive two examples. Consider first the social movement model proposed in [49,50]. In this model, each individual can be in one of three states: member (of the movement), potential member, and ex-member. Individuals interact in a fully-mixed way, with each interaction between a potential member and a member resulting in the potential member becoming a member with probability $\beta'$, and each interaction between a member and an ex-member resulting in the member becoming an ex-member with probability $\delta_1'$; members also "spontaneously" become ex-members with probability $\delta_2'$. The connection between this representation and standard epidemiological models [1] is clear.

Under the assumption of fully-mixed interactions at the meso-scale, standard manipulations yield the following representation for the social dynamics within network communities:

$$\Sigma_H: \qquad \begin{aligned} dP/dt &= -\beta PM-(\beta PM)^{1/2}\eta_1(t), \\ dM/dt &= \beta PM+(\beta PM)^{1/2}\eta_1(t)-\delta_1 ME-(\delta_1 ME)^{1/2}\eta_2(t) -\delta_2 M-(\delta_2 M)^{1/2}\eta_3(t), \\ dE/dt &= \delta_1 ME+(\delta_1 ME)^{1/2}\eta_2(t)+\delta_2 M+(\delta_2 M)^{1/2}\eta_3(t), \end{aligned}$$

where P, M, and E denote the fractions of potential members, members, and ex-members in the community population, $\beta$, $\delta_1$, and $\delta_2$ are nonnegative constants related to the probabilities $\beta'$, $\delta_1'$, and $\delta_2'$ defined above, and the $\eta_i(t)$ are appropriate random processes [e.g., 17]. The deterministic version of this basic model (i.e., with $\eta_1(t)=\eta_2(t)=\eta_3(t)\equiv 0$) is discussed by Hedstrom and coauthors in [49,50], and therefore we denote the model $\Sigma_H$. The deterministic version is shown in [49] to provide a useful description for the *local* growth of a real world social movement.

The second example incorporates the fact that innovations often have both enthusiasts and skeptics, each of whom may actively attempt to recruit the uncommitted. The model $\Sigma_H$ can be modified to account for this competition in recruitment:

$$\Sigma_B: \qquad \begin{aligned} dP/dt &= -\beta_1 PM_1 - \beta_2 PM_2, \\ dM_1/dt &= \beta_1 PM_1 - \delta_1 M_1, \\ dM_2/dt &= \beta_2 PM_2 - \delta_2 M_2, \\ dE/dt &= \delta_1 M_1 + \delta_2 M_2, \end{aligned}$$

where P and E denote the fractions of potential members and ex-members, as before, $M_1$ and $M_2$ are members of the competing groups or movements, and $\beta_1$, $\beta_2$, $\delta_1$, and $\delta_1$ are nonnegative constants. A model of this basic form is proposed in Bettencourt and coworkers in [51] and thus we label it $\Sigma_B$. The model can be fitted, with good agreement, to empirical data for the diffusion of Feynman diagrams (an innovation in physics) in the post World War II era [51]. Developing a stochastic version of $\Sigma_B$, analogous to the representation $\Sigma_H$, is straightforward [39].

The meso-scale model describes the way individual agent decision functions interact to produce collective behavior within social network communities. Individuals also interact with people from other communities, of course, and receive information from channels that transmit to many communities simultaneously (e.g., mass media). These inter-community interactions and "global" social signals are quantified at the macro-scale level of the multi-scale modeling framework. The basic idea is simple and natural: we model interdependence between social network communities with a graph $G_{sc} = \{V_{sc}, E_{sc}\}$, where $V_{sc}$ and $E_{sc}$ are the vertex and edge sets, respectively, $|V_{sc}| = K$, each vertex $v \in V_{sc}$ is a community, and each directed edge $e = (v,v') \in E_{sc}$ represents a potential inter-community interaction. More specifically, an edge $(v,v')$ indicates that an agent in community $v'$ can receive decision-relevant information from one in community v. The way agents act upon this information is specified by their decision functions $r_i$. The broadcast of global social signals to individuals is modeled as a community-dependent input $u_v$ to each individual in community v. Thus $G_{sc}$ and the $u_v$ define the macro-scale model structure.

A key task in deriving a macro-scale model is specifying the topology of $G_{sc}$, as this graph encodes the social network structure for the phenomenon of interest. The most direct approach to constructing $G_{sc}$ is to infer communities directly from social network data, by partitioning the network so as to maximizing the graph modularity $Q_m$. The main challenge with this method for building social community graphs is obtaining the requisite social network data. While this task is certainly nontrivial, availability of such data has increased dramatically over the past decade. For instance, social relationships and interactions increasingly leave "fingerprints" in electronic databases (e.g., communication via email and cell phones, financial transactions), making convenient the acquisition, manipulation, storage, and analysis of these records [e.g., 4].

Alternatively, demographics data can sometimes be used to define both the communities themselves (e.g., families, physical neighborhoods) and their proximity. The basic idea is familiar: individuals belong to social groups, which in turn belong to "groups of groups", and so on, giving rise to a hierarchical organization of communities. For instance, in academics, research groups often belong to academic departments, which are organized into colleges, which in turn form universities, and so on. The proximity of two communities is specified by their relationship within the hierarchy, and this distance defines the likelihood that individuals from the two communities will interact. The probability of inter-community interaction, in turn, can be used to define the network community graph $G_{sc}$ [39].

### A1.2 S-HDS Model Formulation

We now show that the stochastic hybrid dynamical system formalism provides a rigorous, tractable, and expressive framework within which to represent multi-scale social dynamics models. Consider the following

**Definition A1.1:** A *stochastic hybrid dynamical system* (S-HDS) is a feedback interconnection of a continuous-time, continuous state-dependent Markov chain $\{Q, \Lambda(x)\}$ and a collection of stochastic differential equations indexed by the Markov chain state q:

$$\Sigma_{\text{S-HDS}}: \quad \begin{array}{c} \{Q, \Lambda(x)\}, \\ dx = f_q(x,p)dt + G_q(x,p)dw, \end{array}$$

where $q \in Q$ is the discrete state, $x \in X \subseteq \Re^n$ is the continuous state, $p \in \Re^p$ is a vector of system parameters, $\{f_q\}$ and $\{G_q\}$ are sets of vector and matrix fields characterizing the continuous system dynamics, w is an m-valued Weiner process, and $\Lambda(x)$ is the matrix of (x-dependent) Markov chain transition rates; the entries of $\Lambda(x)$ satisfy $\lambda_{qq'}(x) \geq 0$ if $q \neq q'$ and $\Sigma_{q'} \lambda_{qq'}(x) = 0 \ \forall q$, and are related to the standard Markov state transition probabilities as follows [e.g., 34]:

$$P\{q(t+\Delta) = q' | q(t) = q\} = \begin{cases} \lambda_{qq'}(x(t))\Delta + o(\Delta) & \text{if } q \neq q' \\ 1 + \lambda_{qq}(x(t))\Delta + o(\Delta) & \text{if } q = q'. \end{cases}$$

A general discussion of S-HDS theory and applications is beyond the scope of this paper and may be found in, for instance, [34] and the references therein.

We now develop an S-HDS representation for multi-scale social diffusion processes. It is assumed that:

- the social system consists of N individuals distributed over K network communities;
- individuals can influence each other via positive externalities;
- intra-community interactions are fully-mixed;
- inter-community interactions involve the (possibly temporary) migration of individuals from one community to another.

The phenomenon of interest is the diffusion of innovations, in which an innovation of some kind (e.g., a new technology or idea) is introduced into a social system, and individuals may learn about the innovation from others and decide to adopt it [e.g., 2]. By definition an innovation is "new", and therefore it is supposed that initially only a few of the network communities have been exposed to it. An important task in applications is to be able to characterize the likelihood that the innovation will spread to a significant fraction of the population [17].

We model social diffusion as follows:

**Definition A1.2:** The *multi-scale S-HDS diffusion model* is a tuple

$$\Sigma_{\text{S-HDS, diff}} = \{G_{sc}, Q\times X, \{f_q(x), G_q(x), H_q(x)\}_{q\in Q}, \text{Par}, W, U, \{Q, \Lambda(x)\}\}$$

where

- $G_{sc} = \{V_{sc}, E_{sc}\}$ is the social network community graph;
- $Q\times X$ is the system state set, with Q and $X \subseteq \Re^n$ denoting the (finite) discrete and (bounded) continuous state sets, respectively;
- $\{f_q(x), G_q(x), H_q(x)\}_{q\in Q}$, Par, W, U is the S-HDS continuous system, a family of stochastic differential equations which characterizes the intra-community dynamics via vector field/ matrix families $\{f_q\}, \{G_q\}, \{H_q\}$, system parameter vector $p\in \text{Par}\subseteq\Re^p$, and system inputs $w\in W\subseteq\Re^m$, $u\in U\subseteq\Re^r$;
- $\{Q, \Lambda(x)\}$ is the S-HDS discrete system, a continuous-time Markov chain which defines inter-community interactions via state set Q and transition rate matrix $\Lambda(x)$.

The social community graph $G_{sc}$ defines the feasible community-community innovation diffusion pathways: if $(v,v') \notin E_{sc}$ then it is not possible for the innovation to spread directly from community v to community v′. The discrete state set $Q = \{0,1\}^K$ specifies which communities contain at least one adopter of the innovation by labeling such communities with a '1' (and a '0' otherwise). Thus, for example, state $q = [1\ \ 0\ \ 0\ \ \ldots\ ]^T$ indicates that community 1 has at least one adopter, community 2 and 3 do not, and so on. The continuous state space X has coordinates $x_{ij} \in [0,1]$, where $x_{ij}$ is the ith state variable for the continuous system dynamics evolving in community j. For consistency we use the first coordinate for each community, $x_{1j}$, to refer to the fraction of adopters for that community. The continuous system dynamics is defined by a family of q-indexed stochastic differential equations $\{\Sigma_{cs,\,q}\}_{q\in Q}$, with

$$\Sigma_{cs,\,q}: \qquad dx = f_q(x,p)dt + G_q(x,p)dw + H_q(x,p)du,$$

where $w\in W$ is a standard Weiner process and $u\in U$ is the exogenous input. Ordinarily w is interpreted as a stochastic "disturbance", while u is employed to represent influences from "global" sources such as mass media. These dynamics quantify intra-community diffusion of the innovation of interest, for instance through models of the form $\Sigma_H$. The Markov chain matrix $\Lambda(x)$ specifies the transition rates for discrete state transitions $q \rightarrow q'$ and depends on both $G_{sc}$ and x (e.g., the rate at which community v will "infect" other communities depends upon the fraction of adopters in v). It is worth noting that the model $\Sigma_{\text{S-HDS, diff}}$ naturally accommodates both probabilistic (via w and the Markov chain dynamics) and set-bounded (through parameter set Par) uncertainty descriptions, as this expressiveness is desirable in applications.

### A1.3 A Simple Example

We now demonstrate the implementation of the proposed multi-scale S-HDS diffusion modeling framework, and illustrate its efficacy, through a simple example; a more complex example, with more interesting analytic goals, is investigated in Appendix Two below. Consider a social network consisting of two communities and a social movement process playing out on this network. We construct the social network using the method given in [52]. Briefly, a collection of N vertices is divided into two communities of equal size, denoted L and R (for 'left' and 'right', see Figure 6). For all vertex pairs, if both vertices belong to the same community then an edge is placed between them with probability $p_i$, and if the vertices belong to different communities then they are connected with probability $p_e < p_i$. Increasing the ratio $p_i$ / $p_e$ makes the resulting network more "community-like" by increasing the relative intra-community edge density. Figure 6 shows two small example networks built in this way, with the network on the left corresponding to a larger $p_i$ / $p_e$ ratio.

The social movement dynamics evolving on this network is a "network version" of the model proposed in [49]. Thus each individual can be in one of three states – member, potential member, and ex-member – and individuals can change states in one of three ways: 1.) members persuade potential members to whom they are linked to become members with probability $\beta'$, 2.) ex-members likewise influence neighboring members to become ex-members with probability $\delta_1'$, and 3.) members can spontaneously become ex-members with probability $\delta_2'$. For convenience of reference this "agent-based" system representation is denoted $\Sigma_{\text{ABM}}$.

It is straightforward to derive an S-HDS version of the social movement model $\Sigma_{\text{ABM}}$. Consider the diffusion model $\Sigma_{\text{S-HDS, diff}} = \{G_{sc}, Q{\times}X, \{f_q(x), G_q(x), H_q(x)\}_{q \in Q}$, Par, W, U, $\{Q, \Lambda(x)\}\}$ specified in Definition A1.2. Note first that in this case the social network community graph $G_{sc}$ is very simple, consisting of two vertices corresponding to communities L and R and an undirected edge connecting them. The continuous system state is $x = [P_L \;\; M_L \;\; P_R \;\; M_R]^T \in X$, where the subscripts indicate communities (note that the concentrations of ex-members, $E_L$ and $E_R$, are not independent states because the total concentration sums to one on each community). We approximate the agent-based social movement dynamics *within* each network community with the fully-mixed model $\Sigma_H$, that is, with a set of stochastic differential equations governing the evolution of the concentrations of members M and potential members P.

It can be seen that $\Sigma_H$ together with the preceding discussion defines the model components X, $\{f_q(x),G_q(x),H_q(x)\}_{q \in Q}$, Par, W, U that make up the continuous system portion of $\Sigma_{\text{S-HDS, diff}}$. Thus all that remains is to specify the discrete system $\{Q, \Lambda(x)\}$. The discrete state set Q = {00, 10, 01, 11} indicates which communities contain at least one movement member, so that for instance state q = 10 indicates that community L has at least one member and community R has no members. The Markov chain matrix $\Lambda(x)$ specifies the transition rates for discrete state transitions $q \rightarrow q'$. These rates depend on the continuous system state x because the likelihood that one community will "infect" the other depends upon the current concentrations of members, potential members, and ex-members in that community.

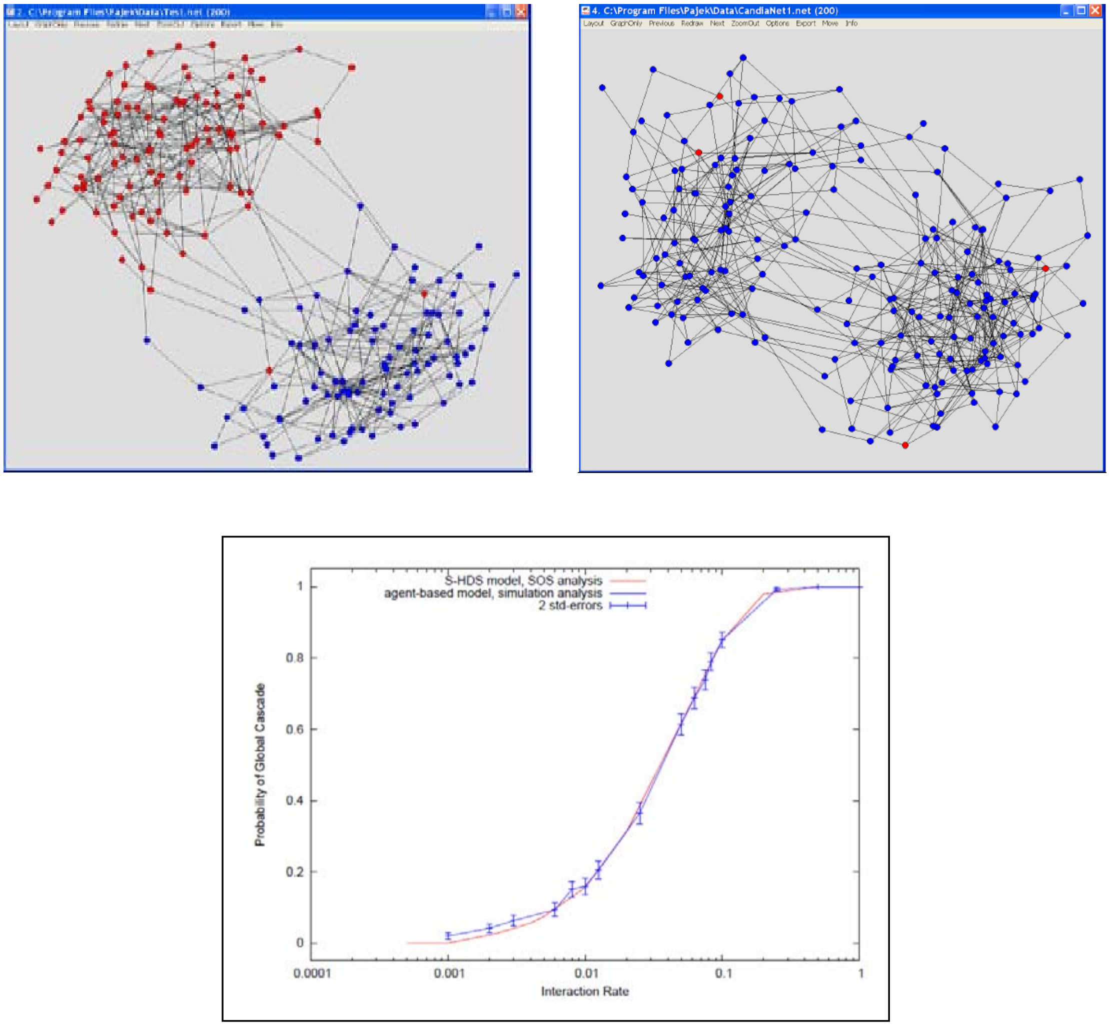


**Figure 6.** Sample results for ABM/S-HDS comparison study. The visualization at top is a cartoon of the basic setup, in which an innovation is introduced into one of the two network communities comprising a social system; possible outcomes include diffusion of the innovation throughout the community initially "infected" (left network, blue vertices are in state M) or across both communities (right network). The plot at bottom shows the probability of "global" diffusion as a function of inter-community interaction intensity for the models $\Sigma_{\text{ABM}}$ (blue curve) and $\Sigma_{\text{S-HDS, diff}}$ (red curve).

We examine the utility of the S-HDS social diffusion model constructed above by using this model to estimate the probability that a small set of "seed" members introduced into community L will lead to the movement growing and eventually propagating to community R. Because the model $\Sigma_{\text{S-HDS, diff}}$ is derived from $\Sigma_{\text{ABM}}$, $\Sigma_{\text{ABM}}$ is taken to be ground truth and $\Sigma_{\text{S-HDS, diff}}$ is deemed a useful approximation if the cascade probability estimates obtained using the S-HDS representation are in good agreement with those computed based on $\Sigma_{\text{ABM}}$. The following parameter values are chosen for $\Sigma_{\text{ABM}}$: N = 2000, $\beta' = 0.5$, $\delta_1' = 0.01$, $\delta_2' = 0.1$ (the results reported are not sensitive to variation in these values). We build 50 random realizations of the social network for each of 15 $p_i$ / $p_e$ ratios. The values for $p_i$ / $p_e$ are selected to generate a collection of 15 network sets whose topologies interpolate smoothly between networks with essentially disconnected communities (large $p_i$ / $p_e$) and networks whose two communities are tightly coupled (small $p_i$ / $p_e$). A "global" cascade is said to occur if an initial seed set of five movement members in community R, chosen at random, results in the diffusion of the movement to community L. The probability of global cascade at a given $p_i$ / $p_e$ ratio is computed by running 20 simulations on each of the 50 social network realizations associated with that $p_i$ / $p_e$, and counting up those for which the innovation propagates to community L. The results of this simulation study are presented in the plot at the bottom of Figure 6, with the blue curve showing the probability estimates as a function of $p_i$ / $p_e$ ratio and the error bars corresponding to ± 2 standard errors.

We now investigate the efficacy of the S-HDS social diffusion model by using this model to estimate the probability of global cascade. The social diffusion model $\Sigma_{\text{S-HDS, diff}}$ is instantiated to be equivalent to the agent-based representation $\Sigma_{\text{ABM}}$ described above. Note that, in particular, there are no free parameters available to permit the response of $\Sigma_{\text{S-HDS, diff}}$ to be "tuned" to match $\Sigma_{\text{ABM}}$. For instance, the $\Sigma_{\text{ABM}}$ parameters $\beta'$, $\delta_1'$, $\delta_2'$ uniquely define $\Sigma_{\text{S-HDS, diff}}$ parameters $\beta$, $\delta_1$, $\delta_2$, and specifying values for the $p_i$ / $p_e$ ratios gives corresponding values for the S-HDS transition matrices $\Lambda(x)$ (to within a single "offset" parameter, see [39]). A Matlab program implementing the resulting model $\Sigma_{\text{S-HDS, diff}}$ is given in [39].

In order to compute the probability of global cascade using the S-HDS model $\Sigma_{\text{S-HDS, diff}}$, we employ the "altitude function" method described in Appendix Two below. This method calculates provably-correct upper bounds on the probability of the social movement propagating to community L. The results of this analysis are given at the plot of the bottom of Figure 6 (red curve). Observe that the global cascade probability estimates obtained using the two models $\Sigma_{\text{ABM}}$ and $\Sigma_{\text{S-HDS, diff}}$ are in close agreement. As it is challenging to model "discontinuous" phenomena such as diffusion across social network communities, this agreement represents important evidence that the S-HDS provides a useful characterization of social diffusion on networks.

While the models $\Sigma_{\text{ABM}}$ and $\Sigma_{\text{S-HDS, diff}}$ generate similar results in this example, the S-HDS representation is much more efficient computationally. For instance, estimating the desired global cascade probabilities using the S-HDS model requires less than one percent of the computer time needed to obtain these estimates with the equivalent agent-based model. Moreover, this difference on efficiency increases with network size, which is important because realistic social networks have hundreds or thousands of communities rather than just two. This computational tractability hints at a more general, and more significant, mathematical tractability enjoyed by the S-HDS framework, a property we now leverage to develop a rigorous predictive analysis methodology for social diffusion events.

## A2. Appendix Two: Predictive Analysis

In this Appendix we formulate the predictive analysis problem in terms of reachability assessment, show that these reachability questions can be addressed through an "altitude function" analysis *without computing system trajectories*, and apply this theoretical framework to demonstrate that predictability of a broad class of social diffusion models depends crucially upon the meso-scale topological structures of the underlying networks. For convenience of exposition, in this Appendix we focus on network communities as a representative meso-scale structure; however, all results derived here are also applicable to the more general case in which the "network partition" (see Section 2.2) includes both community and core-periphery structures.

### A2.1 Predictive Analysis as Reachability Assessment

We propose that accurate prediction requires careful consideration of the interplay between the intrinsics of a process and the social dynamics which are its realization. We therefore adopt an inherently dynamical approach to predictive analysis: given a social process, a set of measurables, and the behavior of interest, we formulate prediction problems as questions about the reachability properties of the system. Toward that end, the behavior about which predictions are to be made is used to define the system *state space subsets of interest* (SSI), while the particular set of candidate measurables under consideration allows identification of the *candidate starting set* (CSS), that is, the set of states and system parameter values which represent initializations that are equivalent under the assumed observational capability. This setup permits predictability assessment, and the related task of identifying useful measurables, to be performed in a systematic manner. Roughly speaking, the proposed approach to predictability assessment involves determining how probable it is to reach the SSI from a CSS and deciding if these reachability properties are compatible with the prediction goals. If a system's reachability characteristics are incompatible with the given prediction question – if, say, "hit" and "flop" in a cultural market are both likely to be reached from the CSS – then the prediction objectives should be refined in some way. Possible refinements include relaxing the level of detail to be predicted or introducing additional measurables.

We now make these notions more precise. Consider the multi-scale S-HDS social diffusion model $\Sigma_{\text{S-HDS, diff}}$ specified in Definition A1.2. Let $P_0$ be a subset of the parameter set Par and $X_0$, $X_{s1}$, $X_{s2}$ be subsets of the (bounded) continuous system state space X. Suppose $X_0 \times P_0$ and $\{X_{s1}, X_{s2}\}$ are the CSS and SSI, respectively, corresponding to the prediction question. Let a specification $\delta > 0$ be given for the minimum acceptable level of variation in system behavior relative to $\{X_{s1}, X_{s2}\}$. Consider the following

**Definition A2.1:** A situation is *eventual state (ES) predictable* if $|\gamma_1 - \gamma_2| > \delta$, where $\gamma_1$ and $\gamma_2$ are the probabilities of $\Sigma_{\text{S-HDS, diff}}$ reaching $X_{s1}$ and $X_{s2}$, respectively, and is *ES unpredictable* otherwise.

Note that in ES predictability problems it is expected that the two sets $\{X_{s1}, X_{s2}\}$ represent qualitatively different system behaviors (e.g., hit and flop in a cultural market), so that if the probabilities of reaching each from $X_0 \times P_0$ are similar then system behavior is unpredictable in a sense that is meaningful for many applications. Other useful forms of predictability are defined and investigated in [39].

The notion of predictability forms the basis for our definition of useful measurables:

**Definition A2.2:** Let the components of the vectors $(x_0, p_0) \in X_0 \times P_0$ which comprise the CSS be denoted $x_0 = [x_{01} \dots x_{0n}]^T$ and $p_0 = [p_{01} \dots p_{0p}]^T$. The *measurables with most predictive power* are those state variables $x_{0j}$ and/or parameters $p_{0k}$ for which predictability is most sensitive.

Intuitively, those measurables for which predictability is most sensitive are likely to be the ones that can most dramatically affect the predictability of a given problem. Note that we do not specify a particular measure of sensitivity to be used when identifying measurables with maximum predictive power, as such considerations are ordinarily application-dependent (see [39] for some useful specifications). Definitions A2.1 and A2.2 focus on the role played by *initial* states in the predictability of social processes. In some cases it is useful to expand this formulation to allow consideration of states other than initial states. For instance, we show in [18] that very early time series are often predictive for PEP, suggesting that it can be valuable to consider initial state *trajectory segments*, rather than just initial states, when assessing predictability. This extension can be naturally accomplished by redefining the CSS, for instance by augmenting the state space X with an explicit time coordinate [18].

We now turn our attention to the "early warning" problem.

**Definition A2.3:** Let the event of interest be specified in terms of $\Sigma_{\text{S-HDS, diff}}$ reaching or escaping some SSI $X_s$, and suppose a warning signal is to be issued only if the probability of event occurrence exceeds some specified threshold $\alpha$. *Reach warning analysis* involves identifying a state set $X_w$, where $X_s \subseteq X_w$ necessarily, with the property that if the system trajectory enters $X_w$ then the probability that $\Sigma_{\text{S-HDS, diff}}$ will eventually reach $X_s$ is at least $\alpha$. Analogously, *escape warning analysis* involves identifying a state set $X_w$, where $X \setminus X_w \subseteq X_s$ necessarily, with the property that if the system trajectory enters $X_w$ then the probability that $\Sigma_{\text{S-HDS, diff}}$ will eventually escape from $X_s$ is at least $\alpha$.

### A2.2 Stochastic Reachability Assessment

The previous section formulates predictive analysis problems as reachability questions. Here we show that these reachability questions can be addressed through an "altitude function" analysis, in which we seek a scalar function of the system state that permits conclusions to be made regarding reachability *without computing system trajectories*. We refer to these as altitude functions to provide an intuitive sense of their analytic role: if some measure of "altitude" is low on the CSS and high on an SSI, and if the expected rate of change of altitude along system trajectories is nonincreasing, then it is unlikely for trajectories to reach this SSI from the CSS.

Consider the S-HDS social diffusion model $\Sigma_{\text{S-HDS, diff}}$ evolving on a bounded state space Q × X. We quantify the uncertainty associated with $\Sigma_{\text{S-HDS, diff}}$ by specifying bounds on the possible values for some system parameters and perturbations and giving probabilistic descriptions for other uncertain system elements and disturbances. Given this representation, it is natural to seek a probabilistic assessment of system reachability.

We begin with an investigation of probabilistic reachability on *infinite* time horizons. The following "supermartingale lemma" is proved in [53] and is instrumental in our development:

**Lemma SM:** Consider a stochastic process $\Sigma_s$ with bounded state space X, and let $\underline{x}(t)$ denote the "stopped" process associated with $\Sigma_s$ (i.e., $\underline{x}(t)$ is the trajectory of $\Sigma_s$ which starts at $x_0$ and is stopped if it encounters the boundary of X). If $A(\underline{x}(t))$ is a nonnegative supermartingale then for any $x_0$ and $\lambda > 0$

$$P\{\sup A(\underline{x}(t)) \geq \lambda \mid \underline{x}(0) = x_0\} \leq A(x_0) / \lambda.$$

Denote by $X_0 \subseteq X$ and $X_s \subseteq X$ the initial state set and SSI, respectively, for the continuous system component of $\Sigma_{\text{S-HDS, diff}}$, and assume that X and the parameter set $\text{Par} \subseteq \Re^p$ are both bounded. Thus, for instance, the SSI is a subset of the continuous system state space X alone; this is typically the case in applications and is easily extended if necessary. We are now in a position to state our first stochastic reachability result:

**Theorem 2:** $\gamma$ is an upper bound on the probability of trajectories of $\Sigma_{\text{S-HDS, diff}}$ reaching $X_s$ from $X_0$, while remaining in Q × X, if there is a family of differentiable functions $\{A_q(x)\}_{q \in Q}$ such that

- $A_q(x) \leq \gamma \; \forall x \in X_0, \forall q \in Q$;

- $A_q(x) \geq 1\ \forall x \in X_s,\ \forall q \in Q$;
- $A_q(x) \geq 0\ \forall x \in X,\ \forall q \in Q$;
- $(\partial A_q/\partial x)\ (f_q + H_q\ u) + (1/2)\ \mathrm{tr}\ [G_q^T\ (\partial^2 A_q/\partial x^2)\ G_q] + \Sigma_{q' \in Q}\ \lambda_{qq'}\ A_{q'} \leq 0\ \forall x \in X,\ \forall q \in Q,\ \forall u \in U,\ \forall p \in Par$.

**Proof:** Note first that $BA_q(x) = (\partial A_q/\partial x)\ (f_q + H_q\ u) + (1/2)\ \mathrm{tr}\ [G_q^T\ (\partial^2 A_q/\partial x^2)\ G_q] + \Sigma_{q' \in Q}\ \lambda_{qq'}\ A_{q'}$ is the infinitesimal generator for $\Sigma_{\text{S-HDS, diff}}$, and therefore quantifies the evolution of the expectation of $A_q(x)$ [53,34]. As a consequence, the third and fourth conditions of the theorem imply that $A(q(t),x(t))$ is a nonnegative supermartingale [53]. Thus, from Lemma SM, we can conclude that $P\{x(t) \in X_s$ for some $t\} \leq P\{\sup A(q(t),x(t)) \geq 1 \mid x(0)=x_0\} \leq A(q,x_0) \leq \gamma\ \forall x_0 \in X_0,\ \forall q \in Q,\ \forall u \in U,\ \forall p \in Par$. ■

The preceding result characterizes reachability of S-HDS on infinite time horizons. In some situations, including important applications involving social systems, it is of interest to study system behavior on *finite* time horizons. The following result is useful for such analysis:

**Theorem 3:** $\gamma$ is an upper bound on the probability of trajectories of $\Sigma_{\text{S-HDS, diff}}$ reaching $X_s$ from $X_0$ during time interval $[0,T]$, while remaining in $Q \times X$, if there exists a family of differentiable functions $\{A_q(x,t)\}_{q \in Q}$ such that

- $A_q(x,t) \leq \gamma\ \forall (x,t) \in X_0 \times 0,\ \forall q \in Q$;
- $A_q(x,t) \geq 1\ \forall (x,t) \in X_s \times [0,T],\ \forall q \in Q$;
- $A_q(x,t) \geq 0\ \forall (x,t) \in X \times \Re^+,\ \forall q \in Q$;
- $BA_q(x,t) \leq 0\ \forall (x,t) \in X \times \Re^+,\ \forall q \in Q,\ \forall u \in U,\ \forall p \in Par$.

**Proof:** The proof follows immediately from that of Theorem 2 once it is observed that $P\{\underline{x}(t) \in X_s$ for some $t \in [0,T]\} = P\{(\underline{x}(t),t) \in X_s \times [0,T]\}$. ■

The idea for the proof of Theorem 3 was suggested in [54].

Having formulated predictability assessment for social processes in terms of system reachability and presented a new theoretical methodology for assessing reachability, we are now in a position to give our approach to deciding predictability. Observe first that Theorems 2 and 3 are of direct practical interest only if it is possible to efficiently compute a tight probability bound $\gamma$ and associated altitude function $A(x)$ which satisfy the theorem conditions. Toward that end, observe that the theorems specify *convex* conditions to be satisfied by altitude functions: if $A_1$ and $A_2$ satisfy the theorem conditions then any convex combination of $A_1$ and $A_2$ will also satisfy the conditions. Thus the search for altitude functions can be formulated as a convex programming problem [55]. Moreover, if the system of interest admits a polynomial description (e.g., the system vector and matrix fields are polynomials) and we search to polynomial altitude functions, then the search can be carried out using sum-of-squares (SOS) optimization [56].

SOS optimization is a convex relaxation framework based on SOS decomposition of the relevant polynomials and semidefinite programming. SOS relaxation involves replacing the nonnegative and nonpositive conditions to be satisfied by the altitude functions with SOS conditions. For example, the conditions for $A_q(x)$ given in Theorem 2 can be relaxed as follows:

$$\begin{array}{rcl} A(x) \leq \gamma\ \forall x \in X_0 & \rightarrow & \gamma - A(x) - \lambda_0^T(x)\ g_0(x) \text{ is SOS} \\ A(x) \geq 1\ \forall x \in X_s & \rightarrow & A(x) - 1 - \lambda_s^T(x)\ g_s(x) \text{ is SOS} \\ A(x) \geq 0\ \forall x \in X & \rightarrow & A(x) - \lambda_{X1}^T(x)\ g_{X1}(x) \text{ is SOS} \\ BA(x) \leq 0\ \forall x \in X,\ \forall p \in Par & \rightarrow & -BA(x) - \lambda_{X2}^T(x)\ g_{X2}(x) - \lambda_P^T(p)\ g_P(p) \text{ is SOS} \end{array}$$

where the entries of the vector functions $\lambda_0$, $\lambda_s$, $\lambda_{X1}$, $\lambda_{X2}$, $\lambda_P$ are SOS, the vector functions $g_0$, $g_s$, $g_{X1}$, $g_{X2}$, $g_P$ satisfy $g_*(\cdot) \geq 0$ (entry-wise) whenever $x \in X_*$ or $p \in Par$, respectively, and we assume $|Q| = 1$ for notational convenience. The conditions on $A_q(x,t)$ specified in Theorem 3 can be relaxed in exactly the same manner. The relaxed SOS conditions are clearly sufficient and in practice are typically not overly-conservative [56,39].

Once the set of conditions to be satisfied by A(x) are relaxed in this way, SOS programming can be used to compute $\gamma_{min}$, the minimum value for the probability bound $\gamma$, and A(x), the associated altitude function which certifies the correctness of this bound. Software for solving SOS programs is available as the third-party Matlab toolbox SOSTOOLS [56], and example SOS programs are given in [39]. Importantly, the approach is tractable: for fixed polynomial degrees, the computational complexity of the associated SOS program grows polynomially in the dimension of the continuous state space, the cardinality of the discrete state set, and the dimension of the parameter space.

For completeness, we outline an algorithm for computing the pair ($\gamma_{min}$, A(x)):

**Algorithm A2.1: altitude functions via SOS programming (outline)**

1. Parameterize A as $A(x) = \Sigma_k c_k a_k(x)$, where $\{a_1, \ldots, a_B\}$ are monomials up to a desired degree bound and $\{c_1, \ldots, c_B\}$ are to-be-determined coefficients.
2. Relax all A(x) criteria in the relevant theorem to SOS conditions.
3. Formulate an SOS program with decision variables $\gamma$, $\{c_1, \ldots, c_B\}$, where the desired bound on altitude function polynomial degree is reflected in the specification of the set $\{c_1, \ldots, c_B\}$. Compute the minimum probability bound $\gamma_{min}$ and values for the coefficients $\{c_1, \ldots, c_B\}$ that define A(x) using SOSTOOLS.

It is emphasized that, although the computation of ($\gamma_{min}$, A(x)) is performed numerically, the resulting function A(x) is guaranteed to satisfy the conditions of the relevant theorem and therefore represents a proof of the correctness of the probability upper bound $\gamma_{min}$. Note also that the probability estimate is obtained without computing system trajectories, and is valid for entire sets of initial states $X_0$, parameter values Par, and exogenous inputs U.

Having given a method for efficiently computing pairs ($\gamma_{min}$, A(x)), and thereby characterizing reachability, we are now in a position to sketch an algorithm for assessing ES predictability:

**Algorithm A2.2: ES predictability (outline)**

Given: social diffusion process of interest is $\Sigma_{\text{S-HDS, diff}}$, CSS = $X_0 \times P_0$, SSI = $\{X_{s1}, X_{s2}\}$, and minimum acceptable level of variation = $\delta$.

Procedure:

1. compute (upper bound for) probability $\gamma_1$ of $\Sigma_{\text{S-HDS, diff}}$ reaching $X_{s1}$ from $X_0 \times P_0$;
2. compute (upper bound for) probability $\gamma_2$ of $\Sigma_{\text{S-HDS, diff}}$ reaching $X_{s2}$ from $X_0 \times P_0$;
3. if $|\gamma_1 - \gamma_2| > \delta$ then problem is ES predictable, else problem is ES unpredictable.

Note: $\gamma_1$, $\gamma_2$ can be computed using Theorem 2 (infinite time horizon) or Theorem 3 (finite time horizon) together with Algorithm3.1 and SOSTOOLS [56].

### A2.3 Application to Social Diffusion

The theoretical framework developed in the preceding sections is now used, in combination with empirically-grounded models for social diffusion [e.g., 17,49-51], to demonstrate that predictability of this class of diffusion models depends crucially upon network community structure. We investigate the following predictability question: Is the diffusion of social movements and mobilizations ES predictable and, if so, which measurable quantities have predictive power?

We adopt a specific version of the S-HDS social diffusion model proposed in Definition 2.2:

$$\Sigma_{\text{S-HDS, diff}} = \{G_{sc}, Q\times X, \{f_q(x),G_q(x)\}_{q\in Q}, \text{Par}, W, \{Q, \Lambda(x)\}\}$$

where

- the social network community graph $G_{sc}$ consists of K communities (so $|V_{sc}| = K$), connected together with an Erdos-Renyi random graph topology, with community size drawn from a power law distribution [36];
- each continuous system $\Sigma_{cs, q}$: $dx = f_q(x,p)dt + G_q(x,p)dw$, $q\in Q$, is given by the meso-scale social movement model $\Sigma_H$ or $\Sigma_B$ with appropriate parameter vector p and system "noise" w;
- the discrete system $\{Q, \Lambda(x)\}$ is a Markov chain that defines inter-community interactions in the manner described in Definition A1.2.

A Matlab instantiation of this S-HDS diffusion model is given in [39] and is available upon request. The behavior of the model can be shown to be consistent with empirical observations of several historical social movements (e.g., various movements in Sweden) [39].

In order to assess ES predictability, SSI = $\{X_{s1}, X_{s2}\}$ is defined so that $X_{s1}$, $X_{s2}$ are state sets corresponding to *global* (affecting a significant fraction of the population) and *local* (remaining confined to a small fraction of the population) movement events, respectively. We then employ Algorithm A2.2 iteratively to search for a definition for CSS = $X_0 \times P_0$ which ensures that the probabilities of reaching $X_{s1}$ and $X_{s2}$ from $X_0 \times P_0$ are sufficiently different to yield an ES predictable situation. We use two models of the form $\Sigma_{\text{S-HDS, diff}}$ for this analysis, corresponding to the two definitions for the continuous system $\Sigma_H$ and $\Sigma_B$. Each model is composed of K = 10 communities connected together with an Erdos-Renyi random graph topology. (Using different realizations of the Erdos-Renyi random graph does not affect the conclusions reported below.)

ES predictability analysis yields two main results. First, both the intra-community and inter-community dynamics exhibit *threshold* behavior: small changes in either the intra-community "infectivity" or inter-community interaction rate around their threshold values lead to large variations in the probability that the movement will propagate "globally". More quantitatively, for the diffusion model $\Sigma_{\text{S-HDS, diff}}$ with continuous system dynamics $\Sigma_H$, threshold behavior is obtained when varying 1.) the generalized reproduction number $R = \beta / \delta_2$ and 2.) the rate $\lambda$ at which inter-community interactions between individuals take place. Thus in order for a social movement to propagate to a significant fraction of the population, the threshold conditions $R\geq 1$ and $\lambda\geq\lambda_0$ must be satisfied simultaneously. An analogous conclusion holds when $\Sigma_H$ is replaced with the diffusion model $\Sigma_B$ in the S-HDS representation. This finding is reminiscent of and extends well-known results for epidemic thresholds in disease propagation models [1].

This threshold behavior is illustrated in the plot at the top right of Figure 7, which shows the way probability of global propagation increases with inter-community interaction rate when the intra-community diffusion is sufficiently infective (i.e., $R\geq 1$). The probabilities which make up this plot represents provably-correct (upper bound) estimates computed using Theorem 2 and Algorithm A2.1. A similar threshold response is observed when varying intra-community infectivity R, provided the inter-community interaction rate satisfies $\lambda\geq\lambda_0$. Importantly, the inter-community interaction threshold $\lambda_0$ is seen to be quite small, indicating that even a few links between network communities enables rapid diffusion of the movement to otherwise disparate regions of the social network. This result suggests that a useful predictor of movement activity in a given community is the level of movement activity among that community's neighbors in $G_{sc}$.

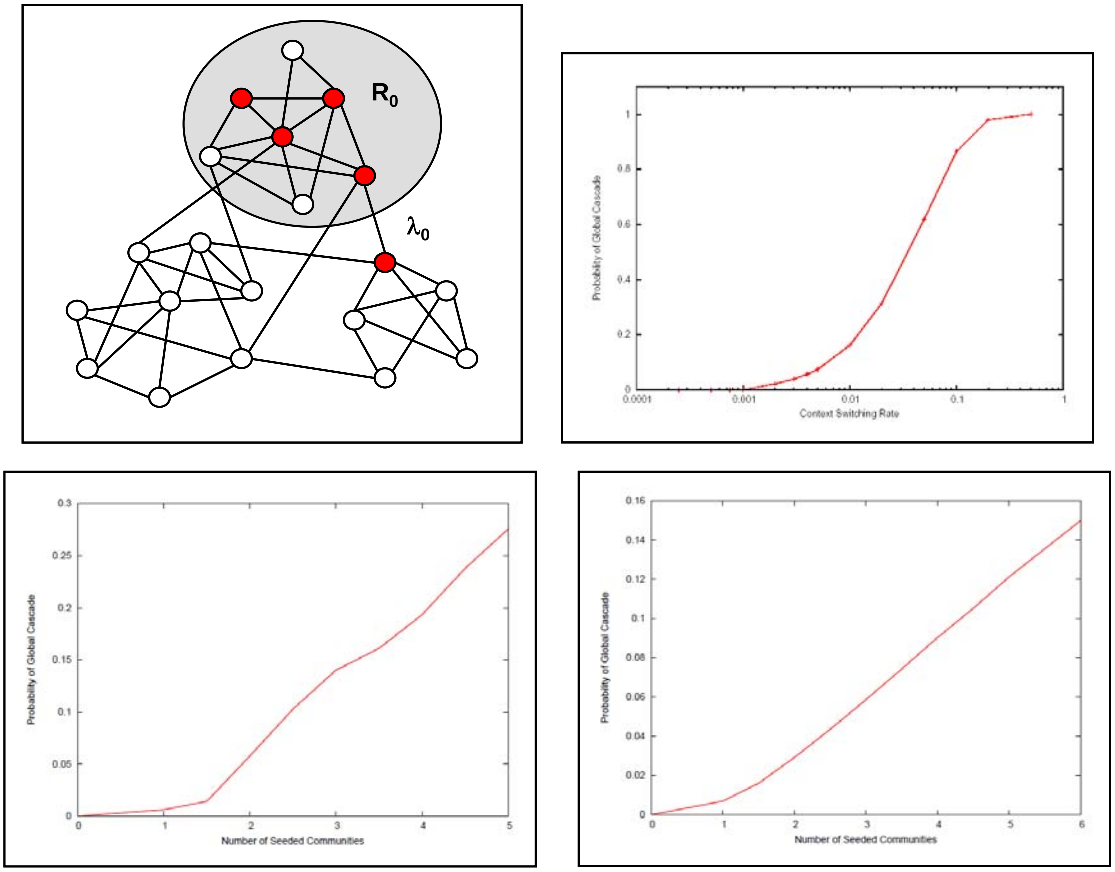


**Figure 7.** Sample results from social diffusion predictability study. Cartoon at top left illustrates the setup for the inter-community interaction study, highlighting the parameter values $R_0$=1 and $\lambda_0$ which quantify intra-and inter-community propagation thresholds; plot at top right shows classic threshold dependence of global propagation probability on inter-community interaction intensity $\lambda$. Plots in bottom row depict the way global propagation probability increases with the number of communities across which a fixed set of innovating seeds are distributed (plots at left and right show cascade probabilities for multi-scale models possessing $\Sigma_H$ and $\Sigma_B$ meso-scale dynamics, respectively).

The second main ES predictability result characterizes the way probability of global propagation varies with the number of network communities across which a *fixed* set of "seed" movement members is distributed. To quantify this dependence, the social movement model $\Sigma_{\text{S-HDS, diff}}$ is initialized so that a small fraction of individuals in the population are movement members and the remainder of the population consists solely of potential members. We then vary the way this initial seed set of movement members is distributed across the K network communities, at one extreme assigning all seeds to the same community and at the other spreading the seeds uniformly over all K communities. For each distribution of seed movement members, the probability of global movement propagation is computed using Theorem 2 and Algorithm A2.1. Other than initialization strategy, the model is specified exactly as in the preceding analysis.

The results of this portion of the ES predictability assessment are summarized in the two plots at the bottom of Figure 7. It is seen that for both choices of meso-scale social movement dynamics, $\Sigma_H$ and $\Sigma_B$, the probability of global movement propagation increases approximately linearly with the number of network communities across which the fixed set of seed members is distributed (here the number of initial members is set to one percent of the total population).

# 1

# Proactive Cyber Defense

There is great interest to develop proactive approaches to cyber defense, in which future attack strategies are anticipated and these insights are incorporated into defense designs. This chapter considers the problem of protecting computer networks against intrusions and other disruptions in a proactive manner. We begin by leveraging the coevolutionary relationship between attackers and defenders to derive two new *proactive filter-based methods* for network defense. The first of these filters is a bipartite graph-based machine learning algorithm which enables information concerning previous attacks to be "transferred" for application against novel attacks, thereby substantially increasing the rate at which defense systems can successfully respond to new attacks. The second approach involves exploiting basic threat information (obtained from, e.g., network security analysts) to generate "synthetic" attack data for use in learning appropriate defense actions, resulting in network defenses that are effective against both current and (near) future attacks. The utility of these two filter-based methods is demonstrated by showing that they outperform standard techniques for the task of detecting malicious network activity in two publicly-available cyber datasets. We then consider the problem of anticipating and characterizing impending attack events with sufficient specificity and timeliness to enable mitigating defensive actions to be taken, and propose a novel *early warning method* as a solution to this problem. The warning method is based upon the fact that certain classes of attacks require the attackers to coordinate their actions, and exploits signatures of this coordination to provide effective attack warning. The potential of the warning-based approach to cyber defense is illustrated through a case study involving politically-motivated Internet attacks.

## 1.1 Introduction

Rapidly advancing technologies and evolving operational practices and requirements increasingly drive both private and public sector enterprises toward highly interconnected and technologically convergent information networks. Proprietary information processing solutions and stove-piped databases are giving way to unified, integrated systems, thereby dramatically increasing the potential impact of even a single well-planned network intrusion, data theft, or denial-of-service (DoS) attack. It is therefore essential that commercial and government organizations develop network defenses which are able to respond rapidly to, or even foresee, new attack strategies and tactics.

Recognizing these trends and challenges, some cyber security researchers and practitioners are focusing their efforts on developing *proactive* methods of cyber defense, in which future attack strategies are anticipated and these insights are incorporated into defense designs [e.g., 1-5]. However, despite this attention, much remains to be done to place the objective of proactive defense on a rigorous and quantitative foundation. Fundamental issues associated with the dynamics and predictability of the coevolutionary “arms race” between attackers and defenders are yet to be resolved. For instance, although recent work has demonstrated that previous attacker actions and defender responses provide predictive information about future attacker behavior [3-5], not much is known about which measurables have predictive power or how to exploit these to form useful predictions. Moreover, even if these predictability and prediction issues were resolved, it is still an open question how to incorporate such predictive analytics into the design of practically-useful cyber defense systems.

This chapter considers the problem of protecting enterprise-scale computer networks against intrusions and other disruptions. We begin by leveraging the coevolutionary relationship between attackers and defenders to develop two *proactive filter-based methods* for network defense. Each of these methods formulates the filtering task as one of behavior classification, in which innocent and malicious network activities are to be distinguished, and each assumes that only very limited prior information is available regarding exemplar attacks or attack attributes. The first method models the data as a bipartite graph of instances of network activities and the features or attributes that characterize these instances. The bipartite graph data model is used to derive a machine learning algorithm which accurately classifies a given instance as either innocent or malicious based upon its behavioral features. The algorithm enables information concerning previous attacks to be “transferred” for use against novel attacks; crucially, it is assumed that previous attacks are drawn from a distribution of attack instances which is related *but not identical* to that associated with the new malicious behaviors. This transfer learning algorithm offers a simple, effective way to extrapolate attacker behavior into the future, and thus significantly increases the speed with which defense systems can successfully respond to new attacks.

The second classifier-based approach to proactive network defense represents attacker-defender coevolution as a hybrid dynamical system (HDS) [6,7], with the HDS discrete system modeling the “modes” of attack (e.g., types of DoS or data exfiltration procedures) and the HDS continuous system generating particular attack instances corresponding to the attack mode presently “active”. Our algorithm takes as input potential near-future modes of attack, obtained for example from the insights of cyber analysts, and generates synthetic attack data for these modes of malicious activity; these data are then combined with recently observed attacks to train a simple classifier to be effective against both current and (near) future attacks. The utility of these proactive filter-based methods is demonstrated by showing that they outperform standard techniques for the task of distinguishing innocent and malicious network behaviors in analyses of two publicly-available cyber datasets.

An alternative approach to proactive network defense is to consider the problem of anticipating and characterizing impending attack events with enough specificity and lead time to allow mitigating defensive actions to be taken. We also explore this approach in the chapter, proposing a novel *early warning method* as a solution to this problem. The proposed warning method is based upon the fact that certain classes of attacks require the attackers to coordinate their actions, often through social media or other observable channels, and exploits signatures generated by this coordination to provide effective attack warning. Interestingly, the most useful early warning indicator identified in this exploratory study is not one of the standard metrics for social media activity, but instead is a subtle measure of the way attack coordination interacts with the *topology* of relevant online social networks. The potential of the early warning approach to proactive cyber defense is illustrated through a case study involving politically-motivated Internet-scale attacks.

## 1.2 Proactive Filters

In this section we propose two filter-based methods for proactive network defense and demonstrate their utility through analysis of publicly-available computer network security-related datasets.

### *1.2.1 Preliminaries*

We approach the task of protecting computer networks from attack as a classification problem, in which the objective is to distinguish innocent and malicious network activity. Each instance of network activity is represented as a feature vector $x \in \Re^{|F|}$, where entry $x_i$ of x is the value of feature i for instance x and F is the set of instance features or attributes of interest (x may be normalized in various ways [7]). Instances can belong to one of two classes: positive / innocent and negative / malicious; generalizing to more than two classes is straightforward. We wish to learn a vector $c \in \Re^{|F|}$ such that the classifier orient = sign($c^T x$) accurately estimates the class label of behavior x, returning +1 (–1) for innocent (malicious) activity.

Knowledge-based classifiers leverage prior domain information to construct the vector c. One way to obtain such a classifier is to assemble a “lexicon” of innocent / positive features $F^{+}\subseteq F$ and malicious / negative features $F^{-}\subseteq F$, and to set $c_i= +1$ if feature i belongs to $F^{+}$, $c_i= -1$ if i is in $F^{-}$, and $c_i=0$ otherwise; this classifier simply sums the positive and negative feature values in the instance and assigns instance class accordingly. Unfortunately this sort of scheme is unable to improve its performance or adapt to new domains, and consequently is usually not very useful in cyber security applications.

Alternatively, learning-based methods attempt to generate the classifier vector c from examples of innocent and malicious network activity. To obtain a learning classifier, one can begin by assembling a set of $n_l$ *labeled* instances $\{(x_i, d_i)\}$, where $d_i\in\{+1, -1\}$ is the class label for instance i. The vector c is then learned through training with the set $\{(x_i, d_i)\}$, for example by solving the following set of equations for c:

$$[X^T X + \gamma I_{|F|}]\ c = X^T d, \qquad (1)$$

where matrix $X\in\Re^{nl\times|F|}$ has instance feature vectors for rows, $d\in\Re^{nl}$ is the vector of instance labels, $I_{|F|}$ denotes the $|F|\times|F|$ identity matrix, and $\gamma\geq 0$ is a constant; this corresponds to regularized least squares (RLS) learning [8]. Many other learning strategies can be used to compute c [8]. Learning-based classifiers have the potential to improve their performance and adapt to new situations, but realizing these capabilities typically requires that large training sets of labeled attacks be obtained. This latter characteristic represents a significant drawback for cyber security applications, where it is desirable to be able to recognize new attacks given only a few (or even no) examples.

In this section we present two new learning-based approaches to cyber defense which are able to perform well with only very modest levels of prior knowledge regarding the attack classes of interest. The basic idea is to leverage “auxiliary” information which is readily available in cyber security applications. More specifically, the first proposed method is a transfer learning algorithm [e.g., 9] which permits the information present in data from previous attacks to be transferred for implementation against new attacks. The second approach uses prior knowledge concerning attack “modes” to generate synthetic attack data for use in training defense systems, resulting in networks defenses which are effective against both current and (near) future attacks.

### 1.2.2 Algorithm One: Transfer Learning

We begin by deriving a bipartite graph-based transfer learning algorithm for distinguishing innocent and malicious network behaviors, and then demonstrate the algorithm’s effectiveness through a case study using publicly-available network intrusion data obtained from the KDD Cup archive [10]. The basic hypothesis is simple and natural: because attacker / defender behavior coevolves, previous activity should provide some indication of future behavior, and transfer learning is one way to quantify and operationalize this intuition.

### Proposed algorithm

The development of the proposed algorithm begins by modeling the problem data as a bipartite graph $G_b$, in which instances of network activity are connected to their features (see Figure 1.1). It is easy to see that the adjacency matrix A for graph $G_b$ is given by

$$A = \begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix} \qquad (2)$$

where matrix $X \in \Re^{n \times |F|}$ is constructed by stacking the n instance feature vectors as rows, and each '0' is a matrix of zeros. In the proposed algorithm, the bipartite graph model $G_b$ is used to exploit the relationships between instances and features by assuming that, in $G_b$, positive / negative instances will tend to be connected to positive / negative features. Note that, as shown below, the learning algorithm can incorporate both instance labels and feature labels (if available). In the case of the latter it is assumed that the feature labels are used to build vector $w \in \Re^{|F|}$, where the entries of w are set to +1 (innocent), −1 (malicious), or 0 (unknown) according to the polarity of the corresponding features.

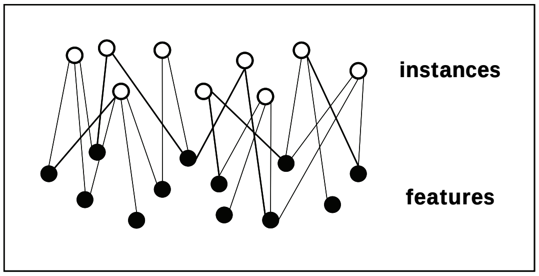


**Fig. 1.1.** Cartoon of bipartite graph model $G_b$. Instances of network activity (white vertices) are connected to the features (black vertices) which characterize them, and link weights (black edges) reflect the magnitudes taken by the features in the associated instances.

Many cyber security applications are characterized by the presence of limited labeled data for the attack class of interest but ample labeled information for a related class of malicious activity. For example, an analyst may be interested in detecting a new class of attacks, and may have in hand a large set of labeled examples of normal network behavior as well as attacks which have been experienced in the recent past. In this setting it is natural to adopt a transfer learning approach, in which knowledge concerning previously observed instances of innocent / malicious behavior, the so-called *source* data, is transferred to permit classification of new *target* data. In what follows we present a new bipartite graph-based approach to transfer learning that is well-suited to cyber defense applications.

Assume that the initial problem data consists of a collection of $n = n_T + n_S$ network events, where $n_T$ is the (small) number of labeled instances available for the target domain, that is, examples of network activity of current interest, and $n_S >> n_T$ is the number of labeled instances from some related source domain, say reflecting recent innocent and malicious activity; suppose also that a modest lexicon $F_l$ of labeled features is known (this set can be empty). Let this label data be used to encode vectors $d_T \in \Re^{nT}$, $d_S \in \Re^{nS}$, and $w \in \Re^{|F|}$, respectively. Denote by $d_{T,est} \in \Re^{nT}$, $d_{S,est} \in \Re^{nS}$, and $c \in \Re^{|F|}$ the vectors of estimated class labels for the target and source instances and the features, and define the *augmented classifier* $c_{aug} = [d_{S,est}^T \quad d_{T,est}^T \quad c^T]^T \in \Re^{n+|F|}$. Note that the quantity $c_{aug}$ is introduced for notational convenience in the subsequent development and is not directly employed for classification.

We derive an algorithm for learning $c_{aug}$, and therefore c, by solving an optimization problem involving the labeled source and target training data, and then use c to estimate the class label of any new instance of network activity via the simple linear classifier orient = sign($c^T x$). This classifier is referred to as *transfer learning-based* because c is learned, in part, by transferring knowledge about the way innocent and malicious network behavior is manifested in a domain which is related to (but need not be identical to) the domain of interest.

We wish to learn an augmented classifier $c_{aug}$ with the following four properties: 1.) if a source instance is labeled, then the corresponding entry of $d_{S,est}$ should be close to this ±1 label; 2.) if a target instance is labeled, then the corresponding entry of $d_{T,est}$ should be close to this ±1 label, and the information encoded in $d_T$ should be emphasized relative to that in the source labels $d_{S,}$; 3.) if a feature is in the lexicon $F_l$, then the corresponding entry of c should be close to this ±1 label; and 4.) if there is an edge $X_{ij}$ of $G_b$ which connects an instance i and a feature j, and $X_{ij}$ possesses significant weight, then the estimated class labels for i and j should be similar.

The four objectives listed above may be realized by solving the following optimization problem:

$$\min_{c_{aug}} \; c_{aug}^T L c_{aug} + \beta_1 \left\| d_{S,est} - k_S d_S \right\|^2 + \beta_2 \left\| d_{T,est} - k_T d_T \right\|^2 + \beta_3 \left\| c - w \right\|^2 \qquad (3)$$

where L = D − A is the graph Laplacian matrix for $G_b$, with D the diagonal degree matrix for A (i.e., $D_{ii} = \Sigma_j A_{ij}$), and $\beta_1$, $\beta_2$, $\beta_3$, $k_S$, and $k_T$ are nonnegative constants. Minimizing (3) enforces the four properties we seek for $c_{aug}$. More specifically, the second, third, and fourth terms penalize “errors” in the first three properties, and choosing $\beta_2 > \beta_1$ and $k_T > k_S$ favors target label data over source labels. To see that the first term enforces the fourth property, note that this expression is a sum of components of the form $X_{ij}$ $(d_{T,est,i} - c_j)^2$ and $X_{ij}$ $(d_{S,est,i} - c_j)^2$. The constants $\beta_1$, $\beta_2$, $\beta_3$ can be used to balance the relative importance of the four properties.

The $c_{aug}$ which minimizes the objective function (3) can be obtained by solving the following set of linear equations:

$$\begin{bmatrix} L_{11} + \beta_1 I_{nS} & L_{12} & L_{13} \\ L_{21} & L_{22} + \beta_2 I_{nT} & L_{23} \\ L_{31} & L_{32} & L_{33} + \beta_3 I_{|F|} \end{bmatrix} c_{aug} = \begin{bmatrix} \beta_1 k_S d_S \\ \beta_2 k_T d_T \\ \beta_3 w \end{bmatrix} \tag{4}$$

where the $L_{ij}$ are matrix blocks of L of appropriate dimension. The system (4) is sparse because the data matrix X is sparse, and therefore large-scale problems can be solved efficiently. Note that in situations where the set of available labeled target instances and features is *very* limited, classifier performance can be improved by replacing L in (4) with the normalized Laplacian $L_n = D^{-1/2} L D^{-1/2}$, or with a power of this matrix $L_n^k$ (for k a positive integer).

We summarize the above discussion by sketching an algorithm for constructing the proposed transfer learning classifier:

*Algorithm TL (Transfer Learning):*

1. Assemble the set of equations (4), possibly by replacing the graph Laplacian L with $L_n^k$.
2. Solve equations (4) for $c_{aug} = [d_{S,est}^T \quad d_{T,est}^T \quad c^T]^T$ (e.g., using the Conjugate Gradient method).
3. Estimate the class label (innocent or malicious) of any new network activity x of interest as: orient = sign($c^T x$).

**Algorithm evaluation**

We now examine the performance of Algorithm TL for the problem of distinguishing innocent and malicious network activity in the KDD Cup 99 dataset, a publicly-available collection of network data consisting of both normal activities and attacks of various kinds [10]. For this study we randomly selected 1000 Normal connections (N), 1000 denial-of-service attacks (DoS), and 1000 unauthorized remote-access events (R2L) to serve as our test data. Additionally, small sets of each of these classes of activity were chosen at random from [10] to be used for training Algorithm TL, and a lexicon of four features, two positive and two negative, was constructed manually and employed to form the lexicon vector w.

We defined two tasks with which to explore the utility of Algorithm TL. In the first, the goal is to distinguish N and DoS instances, and it is assumed that the following data is available to train Algorithm TL: 1.) a set of $d_S/2$ labeled N and $d_S/2$ labeled R2L instances (source data), 2.) a set of $d_T/2$ labeled N and $d_T/2$ labeled

DoS instances (target data), and 3.) the four lexicon features. Thus the source domain consists of N and R2L activities and the target domain is composed of N and DoS instances. In the second task the situation is reversed – the objective is to distinguish N and R2L activities, the source domain is made up of $d_S$ (total) labeled N and DoS instances, and the target domain consists of $d_T$ (total) N and R2L instances. In all tests the number of labeled source instances is $d_S = 50$, while the number of target instances $d_T$ is varied to explore the way classifier performance depends on this key parameter. Of particular interest is determining if it is possible to obtain good performance with only limited target data, as this outcome would suggest both that useful information concerning a given attack class is present in other attacks *and* that Algorithm TL is able to extract this information.

This study compared the classification accuracy of Algorithm TL with that of a well-tuned version of the RLS algorithm (1) and a standard naïve Bayes (NB) algorithm [11]; as the accuracies obtained with the RLS and NB methods were quite similar, we report only the RLS results. Algorithm TL was implemented with the following parameter values: $\beta_1 = 1.0$, $\beta_2 = 3.0$, $\beta_3 = 5.0$, $k_S = 0.5$, $k_T = 1.0$, and $k = 5$. We examined training sets which incorporated the following numbers of target instances: $n_T = 5, 10, 20, 30, 40, 50, 60$. As in previous studies (see, for example, [10]), only the 34 "continuous features" were used for learning the classifiers.

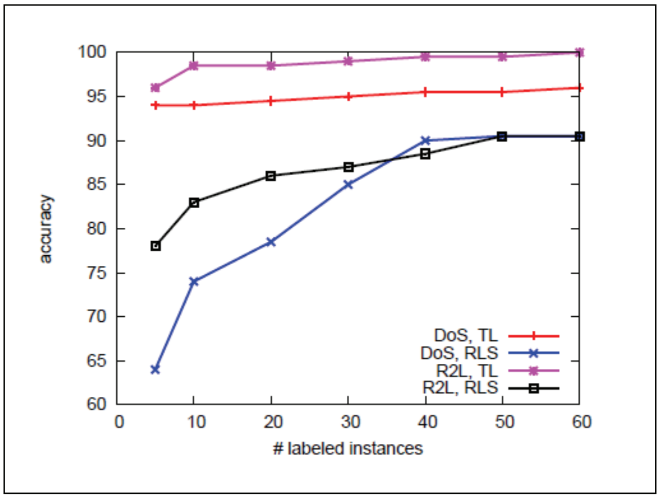


**Fig. 1.2.** Performance of Algorithm TL with limited labeled data. The plot shows how classifier accuracy (vertical axis) varies with the number of available labeled target instances (horizontal axis) for four tasks: distinguish N and DoS using RLS classifier, distinguish N and DoS using Algorithm TL, distinguish N and R2L using RLS classifier, and distinguish N and R2L using Algorithm TL.

Sample results from this study are depicted in Figure 1.2. Each data point in the plots represents the average of 100 trials. It can be seen that Algorithm TL outperforms the RLS classifier (and also the standard NB algorithm, not shown), and that the difference in accuracy of the methods increases substantially as the volume of training data from the target domain becomes small. The performance of Algorithm TL for this task is also superior to that reported for other learning methods tested on these data [e.g., 12]. The ability of Algorithm TL to accurately identify a novel attack after seeing only a very few examples of it, which is a direct consequence of its ability to transfer useful knowledge from related data, is expected to be of considerable value for a range of cyber security applications.

Finally, it is interesting to observe that the bipartite graph formulation of Algorithm TL permits useful information to be extracted from network data *even if no labeled instances are available*. More specifically, we repeated the above study for the case in which $d_T = d_S = 0$, that is, when no labeled instances are available in either the target or source domains. The knowledge reflected in the lexicon vector w is still made available to Algorithm TL. As shown in Figure 1.3, employing a "lexicon only" classifier, in which the vector w is used to build a knowledge-based scheme as described in Section 1.2.1, yields a classification accuracy which is not much better than the 50% baseline achievable with random guessing. However, using this lexicon information together with Algorithm TL enables useful classification accuracy to be obtained (see Figure 1.3). This somewhat surprising result can be explained as follows: the "clustering" property of Algorithm TL encoded in objective function (3) allows the domain knowledge in the lexicon to leverage latent information present in the *unlabeled* target and source instances, thereby boosting classifier accuracy.

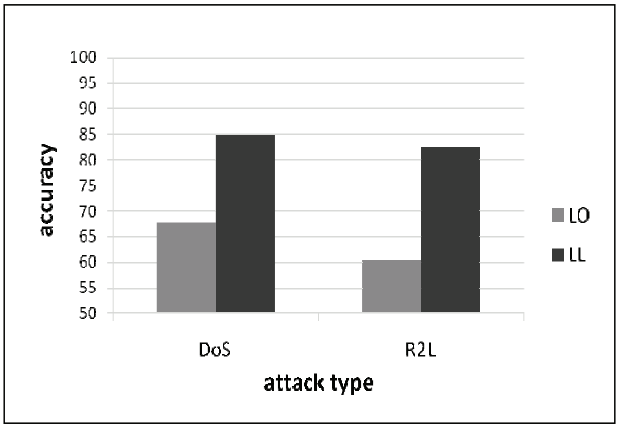


**Fig. 1.3.** Performance of Algorithm TL when no labeled instances are available. The bar graphs depicts classifier accuracy for four tasks: distinguish N and DoS using a lexicon-only (LO) classifier (left, grey bar), distinguish N and DoS using lexicon-learning (LL) via Algorithm TL (left, black bar), distinguish N and R2L using an LO classifier (right, grey bar), and distinguish N and R2L using LL via Algorithm TL (right, black bar).

### 1.2.3 Algorithm Two: Synthetic Attack Generation

In this section we derive our second filter-based algorithm for distinguishing normal and malicious network activity and demonstrate its effectiveness through a case study using the publicly-available Ling-Spam dataset [13]. Again the intuition is that attacker / defender coevolution should make previous activity somewhat indicative of future behavior, and in the present case we operationalize this notion by generating "predicted" attack data and using this synthetic data for classifier training.

**Proposed algorithm**

The development of the second approach to proactive filter-based defense begins by modeling attacker / defender interaction as a stochastic hybrid dynamical system (S-HDS). Here we present a brief, intuitive overview of the basic idea; a comprehensive description of the modeling procedure is given in [7]. An S-HDS (see Figure 1.4) is a feedback interconnection of a discrete-state stochastic process, such as a Markov chain, with a family of continuous-state stochastic dynamical systems [6,14]. Combining discrete and continuous dynamics within a unified, computationally tractable framework offers an expressive, scalable modeling environment that is amenable to formal mathematical analysis. In particular, S-HDS models can be used to efficiently represent and analyze dynamical phenomena which evolve on multiple time scales [14], a property of considerable value in the present application.

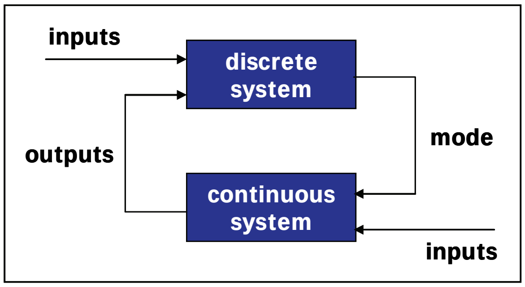


**Fig. 1.4.** Schematic of basic S-HDS feedback structure. The discrete and continuous systems in this framework model the adversary's selection of attack "mode" and resulting attack behavior, respectively, which arise from the coevolving attacker-defender dynamics.

As a simple illustration of the way the S-HDS formalism enables effective, efficient mathematical representation of cyber phenomena, consider the task of modeling the coevolution of Spam attack methods and Spam filters. At an abstract but still useful level, one can think of Spam-Spam filter dynamics as evolving on two timescales:

- the *slow timescale*, which captures the evolution of attack strategies; as an example, consider the way early Spam filters learned to detect Spam by identifying words that were consistently associated with Spam, and how Spammers responded by systematically modifying the wording of their messages, for instance via "add-word" (AW) and "synonym" attacks [15];
- the *fast timescale*, which corresponds to the generation of particular attack instances for a given "mode" of attack (for example, the synthesis of Spam messages according to a specific AW attack method).

We show in [7] that a range of adversarial behavior can be represented within the S-HDS framework, and derive simple but reasonable models for Spam-Spam filter dynamics and for basic classes of network intrusion attacks.

In [14] we develop a mathematically-rigorous procedure for predictive analysis for general classes of S-HDS. Among other capabilities, this analytic methodology enables the predictability of a given dynamics to be assessed and the predictive measurables (if any) to be identified. Applying this predictability assessment process to the adversarial S-HDS models constructed in [7] reveals that, for many of these models, the most predictive measurable is the *mode* of attack, that is, the state variable for the discrete system component of the S-HDS (see [7] for a detailed description of this analysis). Observe that this result is intuitively sensible.

This analytic finding suggests the following *synthetic data learning* (SDL) approach to proactive defense. First, identify the mode(s) of attack of interest. For attacks which are already underway, [7] offers an S-HDS discrete-system state estimation method that allows the mode to be inferred using only modest amounts of measured data. Alternatively, and of more interest in the present application, it is often possible to identify likely future attack modes through analysis of auxiliary information sources (e.g., the subject matter knowledge possessed by domain experts or "non-cyber" data such as that found in social media [16-18]).

Once a candidate attack mode has been identified, synthetic attack data corresponding to the mode can be generated by employing one of the S-HDS models derived in [7]. The synthetic data take the form of a set of K network attack instance vectors, denoted $A_S = \{x_{S1}, \ldots, x_{SK}\}$. The set $A_S$ can then be combined with (actual) measurements of L normal network activity instances, $N_M = \{x_{NM1}, \ldots, x_{NML}\}$, and P (recently) observed attacks, $A_M = \{x_{M1}, \ldots, x_{MP}\}$, yielding the training dataset $TR = N_M \cup A_M \cup A_S$ of real and synthetic data. Note that one effective way to generate a set $A_S$ of synthetic attacks is to use the S-HDS formalism to appropriately *transform* attack instances sampled from the observed attack set $A_M$, rather than to attempt to construct synthetic attacks "from scratch". It is hypothesized that training classifiers with dataset TR may offer a mechanism for deriving defenses which are effective against both current and near future malicious activity.

We summarize the above discussion by sketching a procedure for constructing the proposed SDL classifier:

*Algorithm SDL (Synthetic Data Learning):*

1. Identify the mode(s) of attack of interest (e.g., via domain experts or auxiliary data).
2. Assemble sets of measured normal network activity $N_M$ and measured attack activity $A_M$ for the network under study.
3. Generate a set of synthetic attack instances $A_S$ corresponding to the attack mode(s) identified in Step 1 (for instance by transforming attacks in $A_M$).
4. Train a classifier (e.g., RLS, NB) using the training data $TR = N_M \cup A_M \cup A_S$. Estimate the class label (innocent or malicious) of any new network activity x with the classifier trained using data TR.

### Algorithm evaluation

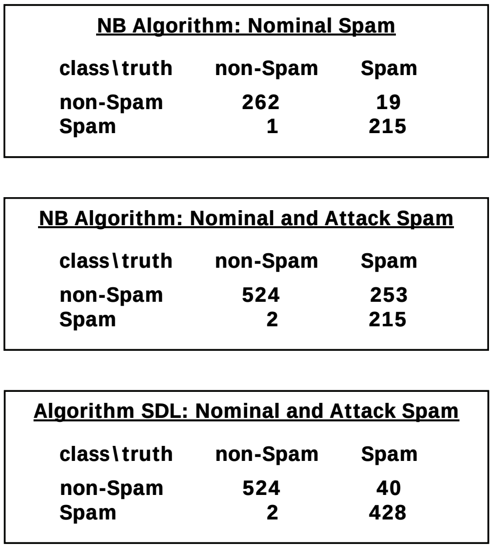
We now examine the performance of Algorithm SDL for the problem of distinguishing legitimate and Spam emails in the Ling-Spam dataset [13], a corpus of 2412 non-Spam emails collected from a linguistics mailing list and 481 Spam emails received by the list. After data cleaning and random sub-sampling of the non-Spam messages we are left with 468 Spam and 526 non-Spam messages for training and testing purposes; this set of 994 emails will be referred to as the *nominal Spam* corpus. (Note that all email was preprocessed using the *ifile* tool [19].)

We considered three scenarios in this study:

1. NB classifier / nominal Spam: for each of ten runs, the nominal Spam corpus was randomly divided into equal-sized training and testing sets and the class label for each message in the test set was estimated with a naïve Bayes (NB) algorithm [11] learned on the training set;
2. NB classifier / nominal plus attack Spam: for each of ten runs, the nominal Spam corpus was randomly divided into equal-sized training and testing sets and the test set was then augmented with 263 additional non-Spam messages (taken from the Ling-Spam dataset) and 234 Spam messages generated via a standard add-word (AW) attack methodology [15]; the class labels for the test messages were estimated with an NB algorithm [11] learned on the nominal Spam training set;
3. Algorithm SDL / nominal plus attack Spam: for each of ten runs, the training and test corpora were constructed exactly as in Scenario 2 and the class labels for the test messages were estimated with Algorithm SDL.

In generating the AW attacks in Scenarios 2 and 3, we assume that the attacker knows to construct AW Spam to defeat an NB filter but does not have knowledge of the specific filter involved [15]. The synthetic AW attacks generated in Scenario 3 (using Step 3 of Algorithm SDL) are computed with no knowledge of the attacker's methodology beyond the mode of attack (i.e., AW).

Sample results from this study are displayed in Figure 1.5. In each case the "confusion matrix" [8] reports the (rounded) average performance over the ten runs. It can be seen that, as expected, the NB filter does well against the nominal Spam but poorly against the AW Spam (in fact, the NB filter does not detect a single instance of AW Spam). In contrast, Algorithm SDL performs well against both nominal Spam and AW Spam, achieving ~96% classification accuracy with a low false positive rate. It is emphasized that this result is obtained using only the (synthetic) estimate of AW Spam generated in Step 3 of Algorithm SDL.



**NB Algorithm: Nominal Spam**

| class\truth | non-Spam | Spam |
|---|---|---|
| non-Spam | 262 | 19 |
| Spam | 1 | 215 |

**NB Algorithm: Nominal and Attack Spam**

| class\truth | non-Spam | Spam |
|---|---|---|
| non-Spam | 524 | 253 |
| Spam | 2 | 215 |

**Algorithm SDL: Nominal and Attack Spam**

| class\truth | non-Spam | Spam |
|---|---|---|
| non-Spam | 524 | 40 |
| Spam | 2 | 428 |

**Fig. 1.5.** Performance of Algorithm SDL on Spam dataset. Each confusion matrix shows number of non-Spam messages classified as non-Spam and Spam (left column) and number of Spam messages classified as non-Spam and Spam (right column). The three matrices, from top to bottom, report the results for: NB against nominal Spam, NB against Spam which contains add-word attacks, and Algorithm SDL against Spam which contains add-word attacks.

## 1.3 Early Warning

In this section we develop an early warning capability for an important class of computer network attacks and illustrate its potential through a case study involving politically-motivated DoS attacks.

### *1.3.1 Preliminaries*

Computer network attacks take many forms, including system compromises, information theft, and denial-of-service attacks intended to disrupt services. In what follows we focus on deriving an early warning capability for distributed denial-of-service (DDoS) attacks, that it, coordinated efforts in which computers are instructed to flood a victim with traffic designed to overwhelm services or consume bandwidth. In particular, we concentrate on politically-motivated DDoS attacks, for three main reasons: 1.) this class of attacks is an important and growing threat [17], 2.) the class is representative of other threats of interest, and 3.) it is expected that in the case of politically-motivated attacks the coordination among attackers may take place, in part, via social media, thereby enabling an analysis employing only publicly-available data.

Consider the task of detecting social media signatures associated with attackers coordinating a politically-motivated DDoS. A classic example of the kind of attack of interest is the sequence of DDoS which were launched against government and commercial sites in Estonia beginning in late April 2007. Interestingly, a retrospective study of these events reveals that there was significant planning and coordination among attackers through web forums and blogs prior to the actual attacks [17], supporting the hypothesis that it may be possible to detect early warning indicators in social media *in advance* of such attacks.

Of course, detecting early warning indicators of an impending DDoS attack in social media is a daunting undertaking. Challenges associated with this task include the vast volume of discussions taking place online, the need to distinguish credible threats from irrelevant chatter, and the necessity to identify reliable attack indicators early enough to be useful (e.g., at least a few days in advance of the attack). Recently we have developed a general framework within which to study this class of early warning problem [14,18,20]. The basic premise is that generating useful predictions about social processes, such as the planning and coordination of a DDoS event, requires careful consideration of the way individuals interact through their social networks. The proposed warning methodology therefore exploits information about social network interactions to forecast which nascent online discussions will ultimately lead to real world attack events, and which will fade into obscurity. Interestingly, the features found to possess exploitable predictive power turn out to be subtle measures of the network dynamics associated with the evolution of early attack-related discussions [14,18,20].

We now briefly summarize the early warning framework presented in [14,18,20] and its application to the DDoS warning problem, and then illustrate the implementation and performance of the warning method through a case study involving politically-motivated Internet attacks.

### 1.3.2 Early Warning Method

In social dynamics, individuals are often affected by what others do. As a consequence, social phenomena can depend upon the topological features of the underlying social network, for instance the degree distribution or presence of small world structure, and aspects of this dependence have been characterized (see [21] for a recent review). We show in [14,18,20] that, for a wide range of social phenomena, useful prediction requires consideration of the way the behavior of individuals interacts with *social network communities*, that is, densely connected groupings of individuals that have only relatively few links to other groups. The concept of network community structure is illustrated in Figure 1.6 and is defined more carefully below. This dependence suggests that in order to derive useful early warning methods for social phenomena, one should consider the topology of the underlying social network; however, standard prediction algorithms do not include such features.

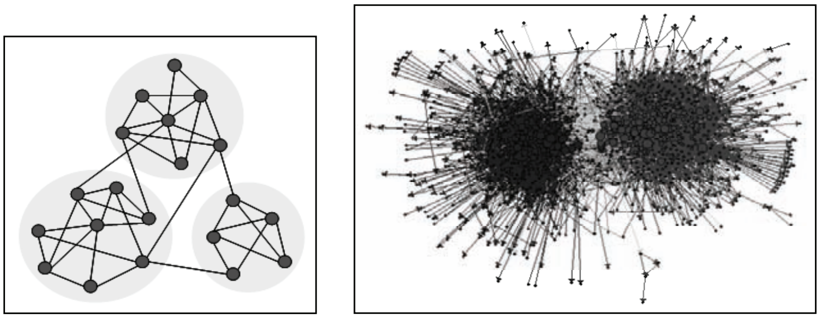


**Fig. 1.6.** Network community structure. Cartoon at left depicts a network with three communities; graph at right is a network of political blogs in which communities of liberal (left cluster) and conservative (right cluster) blogs are clearly visible [22].

While community structure is widely appreciated to be an important topological property in real world social networks, there is not a similar consensus regarding qualitative or quantitative definitions for this concept. Here we adopt the modularity-based definition proposed in [23], whereby a good partitioning of a network's vertices into communities is one for which the number of edges between putative communities is smaller than would be expected in a random partitioning.

To be concrete, a modularity-based partitioning of a network into two communities maximizes the modularity Q, defined as

$$Q = s^T B\, s \,/\, 4m \tag{5}$$

where m is the total number of edges in the network, the partition is specified with the elements of vector s by setting $s_i = 1$ if vertex i belongs to community 1 and $s_i = -1$ if it belongs to community 2, and matrix B has elements $B_{ij} = A_{ij} - k_i k_j / 2m$, with $A_{ij}$ and $k_i$ denoting the network adjacency matrix and degree of vertex i, respectively. Partitions of the network into more than two communities can be constructed recursively [23]. Note that modularity-based community partitions can be efficiently computed for large social networks and require only network topology data for their construction.

Despite the fact that community structure is ubiquitous in real social networks, little has been done to incorporate considerations of communities into social prediction methods. In [14,18,20] we present theoretical and empirical evidence that the predictability of social dynamics often depends crucially upon network community structure. More specifically, we show that early dispersion of a social dynamics "activity" across network communities is a reliable early indicator that the ultimate extent of the activity will be significant. (Perhaps surprisingly, this measure is more predictive than the early magnitude of the activity.)

In the context of early warning for politically-motivated cyber attacks, the social activity of interest is communication associated with planning and coordinating the attack. Thus it is of interest to collect data that enables quantification of the extent to which early communications of this type are dispersed across network communities. Such data should therefore include social network information sufficient to allow the identification of network communities as well as the detection of attack-related discussions among individuals in the network. One way to address this challenge is to adopt *online* social activity as a proxy for real world attack-related discussion and information exchange. More specifically, we use blog posts as our primary data set. The blog network is modeled as a graph in which the vertices are blogs and the edges represent links between blogs, with two blogs being linked if a post in one hyperlinks to a post in the other. Among other things, this blog graph model enables the identification of blog communities: these are the groups of blogs corresponding to the blog graph partition which maximizes the modularity Q for the graph (see (5)); these groups of blogs serve as our proxy for social network communities.

We are now in a position to specify an early warning algorithm for politically-motivated DDoS attacks. The algorithm operationalizes the "early dispersion of attack-related discussions" indicator, computing a measure of the magnitude of this dispersion and issuing an alert if and only if the dispersion is "large".

*Algorithm EW (Early Warning):*

Initialization: Identify a (large) set of cyber security-relevant blogs and forums B to be continually monitored; B should include sites contributed to and frequented by both attackers (e.g., hacker forums) and defenders (e.g., security blogs).

Procedure:

1. Perform *meme detection* with the blogs in B to identify all “memes” which are potentially related to politically-motivated DDoS attacks. Characterize the discussion topic(s) associated with each meme.
2. Conduct a sequence of blog graph crawls and construct a time series of blog graphs $G_B(t)$. For each meme / topic M of interest and each time period t, label the blogs in $G_B(t)$ as ‘active’ if they contain a post containing M and ‘inactive’ otherwise.
3. Form the union $G_B = \cup_t G_B(t)$, partition $G_B$ into network communities, and map the communities structure of $G_B$ back to each of the graphs $G_B(t)$.
4. Compute the post volume time series and the post / community entropy (PCE) time series for each meme / topic.
5. Construct a synthetic ensemble of PCE time series from the post volume dynamics for each meme / topic.
6. Compare the actual PCE time series to the synthetic ensemble series for each meme / topic M to determine if the observed early dispersion of activity across communities is “large” for topic M.

We now offer additional details concerning this procedure; a more comprehensive discussion of the methodology is provided in [7]. Step 1 is performed using the algorithm described in [24,20]. Observe that ‘memes’ in this context are distinctive phrases which propagate relatively unchanged online and act as “tracers” for topics of discussion. It is shown in [24,20] that detecting memes in social media is a useful and general way to discover emerging topics and trends, and we demonstrate in [7] that meme analysis allows the detection of discussions concerning the planning and coordination of politically-motivated DDoS within a day or two of the initiation of these discussions.

Step 2 is by now standard, and various tools exist which can perform these tasks [e.g., 25]. In Step 3, blog network communities are identified with a modularity-based community extraction algorithm applied to the blog graph [23]. In Step 4, the post volume for a given meme / topic M, community i, and sampling interval t is obtained by counting the number of posts containing M made to the blogs comprising community i during interval t. PCE for a particular meme / topic M and sampling interval t is defined as follows:

$$PCE_M(t) = -\Sigma_i f_{M,i}(t) \log(f_{M,i}(t)) \qquad (6)$$

where $f_{M,i}(t)$ is the fraction of total posts containing M and made during interval t which occur in community i. Given the post volume time series obtained in Step 4, Step 5 involves construction of an ensemble of PCE time series that would be expected under “normal circumstances”, that is, if meme M propagated from a small seed set of initiators according to standard models of social diffusion [18,20]. Observe that this step enables us to quantify the expected dispersion for $PCE_M(t)$, so that we can recognize “large” dispersion. Step 6 is carried out by searching for memes M and time periods t during which $PCE_M(t)$ exceeds the mean of the synthetic PCE ensemble by a user-defined threshold (e.g., two standard deviations).

### *1.3.3 Case Study: Politically-Motivated DDoS*

This subsection reports the results of a case study aimed at exploring the ability of Algorithm EW to provide reliable early warning for DDoS attacks. Toward this end, we first identified a set of Internet “disturbances” that included examples from three distinct classes of events:

1. successful politically-motivated DDoS attacks – these are the events for which Algorithm EW is intended to provide warning with sufficient lead time to allow mitigating actions to be taken;
2. natural events which disrupt Internet service – these are disturbances, such as earthquakes and electric power outages, that impact the Internet but for which it is known that no early warning signal exists in social media;
3. quiet periods – these are periods during which there is social media “chatter” concerning impending DDoS attacks but ultimately no (successful) attacks occurred.

Including in the case study events selected from these three classes is intended to provide a fairly comprehensive test of Algorithm EW. For instance, these classes correspond to 1.) the domain of interest (DDoS attacks), 2.) a set of disruptions which impact the Internet but have no social media warning signal, and 3.) a set of “non-events” which do not impact the Internet but do possess putative social media warning signals (discussion of DDoS attacks).

We selected twenty events from these three classes:

Politically-motivated DDoS attacks:

- Estonia event in April 2007;
- CNN / China incident in April 2008;
- Israel / Palestine conflict event in January 2009;
- DDoS associated with Iranian elections in June 2009;
- WikiLeaks event in November 2010;
- Anonymous v. PayPal, etc. attack in December 2010;
- Anonymous v. HBGary attack in February 2011.

Natural disturbances:

- European power outage in November 2006;

- Taiwan earthquake in December 2006;
- Hurricane Ike in September 2008;
- Mediterranean cable cut in January 2009;
- Taiwan earthquake in March 2010;
- Japan earthquake in March 2011.

Quiet periods:

> Seven periods, from March 2005 through March 2011, during which there were discussions in social media of DDoS attacks on various U.S. government agencies but no (successful) attacks occurred.

For brevity, a detailed discussion of these twenty events is not given here; the interested reader is referred to [7] for additional information on these disruptions.

We collected two forms of data for each of the twenty events: *cyber data* and *social data*. The cyber data consist of time series of routing updates which were issued by Internet routers during a one month period surrounding each event. More precisely, these data are the Border Gateway Protocol (BGP) routing updates exchanged between gateway hosts in the Autonomous System network of the Internet. The data was downloaded from the publicly-accessible RIPE collection site [26] using the process described in [27] (see [27] for additional details and background information on BGP routing dynamics). The temporal evolution of the volume of BGP routing updates (e.g., withdrawal messages) gives a coarse-grained measure of the timing and magnitude of large Internet disruptions and thus offers a simple and objective way to characterize the impact of each of the events in our collection. The social data consist of time series of social media mentions of cyber-related memes detected during a one month period surrounding each of the twenty events. These data were collected using the procedure specified in Algorithm EW.

Illustrative time series plots corresponding to two events in the case study, the WikiLeaks DDoS attack in November 2010 and Japan earthquake in March 2011, are shown in Figure 1.7. Observe that the time series of BGP routing updates are similar for the two events, with each experiencing a large “spike” at the time of the event. The time series of blog post volume are also similar across the two events, with each showing modest volume prior to the event and displaying a large spike in activity at event time. However, the time series for blog entropy are quite distinct for the two events. Specifically, in the case of the WikiLeaks DDoS the blog entropy (dashed curve in Figure 1.7) experiences a dramatic increase several days before the event (and leads post volume), while in the case of the Japan earthquake blog entropy is small for the entire collection period (and lags post volume). Similar social media behavior is observed for all events in the case study, suggesting that: 1.) early dispersion of discussions across blog network communities may be a useful early warning indicator for politically-motivated DDoS attacks, and 2.) the post volume associated with these discussions does not appear to be a useful early indicator for these attacks.

To investigate this possibility more carefully, we evaluated the predictive performance of two candidate early warning signals on the twenty events in our test set: 1.) the “early dispersion” PCE indicator computed in Algorithm EW, and 2.) a simple volume-based indicator, in which the presence or absence of significant post volume is used as a signal that a DDoS attack is imminent. We find that the PCE indicator performs well, correctly classifying all twenty events (seven attacks and thirteen non-attacks) and providing an average lead time of sixteen days for attack warning. In contrast, blog volume is not found to be useful for early warning, exhibiting essentially identical behavior for DDoS attacks and natural disturbances and spiking slightly *after* the occurrence of the disruption for all events.

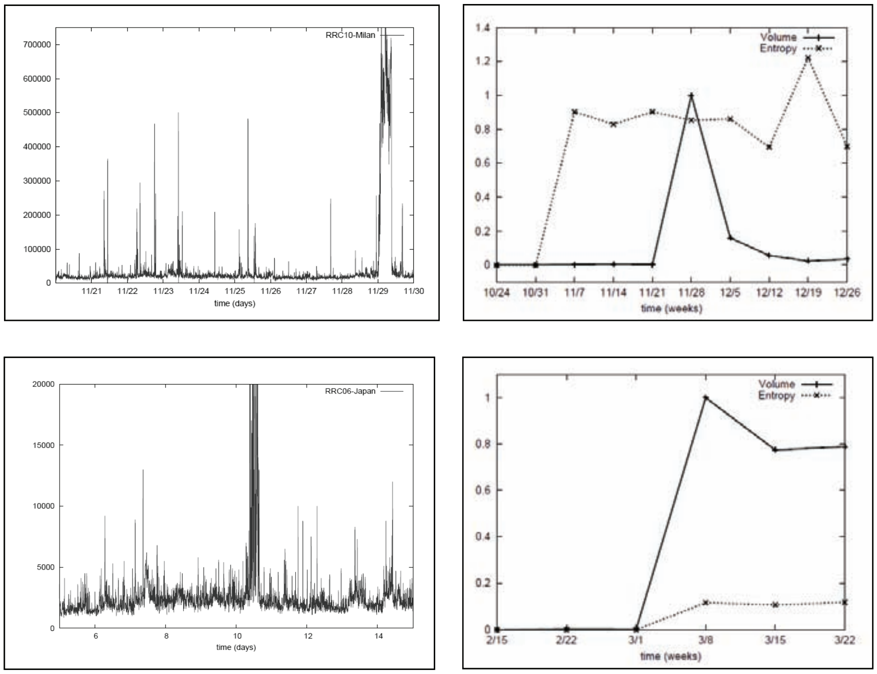


**Fig. 1.7.** Sample results for the DDoS early warning case study. The illustrative time series plots shown correspond to the WikiLeaks event in November 2010 (top row) and the Japan earthquake in March 2011 (bottom row). For each event, the plot at left is the time series of BGP routing updates (note the large increase in updates triggered by the event). The plot at the right of each row is the time series of the social media data, with the solid curve showing blog post volume and the dashed curve depicting blog entropy (in each case, the time series shown are for the meme with largest total volume). Note that while post volume is scaled for convenient visualization, the scale for entropy is consistent across plots to allow cross-event comparison.

## 1.4 Concluding Remarks

This chapter considers the problem of protecting computer networks against intrusions and other disruptions in a proactive manner. We begin by deriving two new proactive filter-based methods for network defense: 1.) a bipartite graph-based transfer learning algorithm which enables information concerning previous attacks to be transferred for application against novel attacks, thereby substantially increasing the rate with which defense systems can successfully respond to new attacks, and 2.) a synthetic data learning method that exploits basic threat information to generate attack data for use in learning appropriate defense actions, resulting in network defenses that are effective against both current and (near) future attacks. The utility of these two filter-based methods is demonstrated by showing that they outperform standard techniques for the task of detecting malicious network activity in two publicly-available cyber datasets. We then present an early warning method as a solution to the problem of anticipating and characterizing impending attack events with sufficient specificity and timeliness to enable mitigating defensive actions to be taken. The warning method is based upon the fact that certain classes of attacks require the attackers to coordinate their actions, and exploits signatures of this coordination to provide effective attack warning. The potential of the warning-based approach to cyber defense is illustrated through a case study involving politically-motivated Internet attacks.

Future work will include application of the proposed proactive defense methods to additional threats, including non-cyber threats which involve attacker-defender coevolution (e.g., counterterrorism), as well as the development of new proactive defense strategies. As an example of one approach toward the latter goal, we have recently shown that adversary activity can be accurately predicted and countered in certain settings by appropriately combining data analysis methods (e.g., machine learning) with behavioral models for adversarial dynamics (e.g., incremental game models) [28].

## Acknowledgements

This work was supported by the Laboratory Directed Research and Development Program at Sandia National Laboratories. We thank Chip Willard of the U.S. Department of Defense for numerous helpful discussions on aspects of this research.

## References

[1] Byers, S. and S. Yang, “Real-time fusion and projection of network intrusion activity”, *Proc. ISIF/IEEE International Conference on Information Fusion,*

Cologne, Germany, July 2008.

[2] Armstrong, R., J. Mayo, and F. Siebenlist, “Complexity science challenges in cybersecurity”, Sandia National Laboratories SAND Report, March 2009.

[3] Colbaugh, R., “Does coevolution in malware adaptation enable predictive analysis?”, *IFA Workshop: Exploring Malware Adaptation Patterns*, San Francisco, CA, May 2010.

[4] Mashevsky, Y., Y. Namestnikov, N. Denishchenko, and P. Zelensky, “Method and system for detection and prediction of computer virus-related epidemics”, US Patent 7,743,419, June 2010.

[5] Bozorgi, M., L. Saul, S. Savage, and G. Voelker, “Beyond heuristics: Learning to classify vulnerabilities and predict exploits”, *Proc. ACM International Conference on Knowledge Discovery and Data Mining*, Washington DC, July 2010.

[6] Majumdar, R. and P. Tabuada, *Hybrid Systems: Computation and Control*, LNCS 5469, Springer, Berlin, 2009.

[7] Colbaugh, R. and K. Glass, “Proactive defense for evolving cyber threats”, Sandia National Laboratories SAND Report, September 2011.

[8] Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Second Edition, Springer, New York, 2009.

[9] Pan, S. and Q. Yang, “A survey on transfer learning”, *IEEE Trans. Knowledge and Data Engineering*, Vol. 22, pp. 1345-1359, 2010.

[10] http://kdd.ics.uci.edu/databases/kddcup99/; last accessed December 2010.

[11] http://www.borgelt.net/bayes.html; last accessed July 2010.

[12] He, J., Y. Liu, and R. Lawrence, “Graph-based transfer learning”, *Proc. ACM Conference on Information and Knowledge Management*, Hong Kong, November 2009.

[13] http://labs-repos.iit.demokritos.gr/skel/i-config/downloads/; last accessed July 2010.

[14] Colbaugh, R. and K. Glass, “Predictive analysis for social processes I: Multiscale hybrid system modeling, and II: Predictability and warning analysis”, *Proc. IEEE International Multi-Conference on Systems and Control*, Saint Petersburg, Russia, July 2009.

[15] Lowd, D. and C. Meeks, “Good word attacks on statistical Spam filters”, *Proc. 2005 Conference on Email and Anti-Spam*, Palo Alto, CA, July 2005.

[16] Cao, L., P. Yu, C. Zhang, H. Zhang, F. Tsai, and K. Chan, “Blog data mining for cyber security threats”, *Data Mining for Business Applications*, Springer US, 2009.

[17] Nazario, J., “Politically motivated denial of service attacks”, in *The Virtual Battlefield: Perspectives on Cyber Warfare*, IOS Press, Amsterdam, 2009.

[18] Colbaugh, R. and K. Glass, “Early warning analysis for social diffusion

events", *Proc. IEEE International Conference on Intelligence and Security Informatics*, Vancouver, Canada, May 2010.

[19] http://www.nongnu.org/ifile/; last accessed July 2010.

[20] Colbaugh, R. and K. Glass, "Emerging topic detection for business intelligence via predictive analysis of 'meme' dynamics", *Proc. AAAI 2011 Spring Symposium*, Palo Alto, CA, March 2011.

[21] Easley, D. and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, 2010.

[22] Adamic, L. and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog", *Proc. ACM International Conference on Knowledge Discovery and Data Mining*, Chicago, August 2005.

[23] Newman, M., "Modularity and community structure in networks", *Proceedings of the National Academy of Sciences USA*, Vol. 103, pp. 8577-8582, 2006.

[24] Leskovec, J., L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle", *Proc. ACM International Conference on Knowledge Discovery and Data Mining*, Paris, France, June 2009.

[25] Glass, K. and R. Colbaugh, "Web analytics for security informatics", *Proc. European Intelligence and Security Informatics Conference*, Athens, Greece, September 2011.

[26] http://data.ris.ripe.net/; last accessed July 2011.

[27] Glass, K., R. Colbaugh, and M. Planck, "Automatically identifying the sources of large Internet events", *Proc. IEEE International Conference on Intelligence and Security Informatics*, Vancouver, Canada, May 2010.

[28] Colbaugh, R., "Monsoons, movies, memes, and genes: Combining KD and M&S for prediction", Keynote Talk, KDMS Workshop, *ACM International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, August 2011.

# Predictability-Oriented Defense Against Adaptive Adversaries

Richard Colbaugh
Sandia National Laboratories
Albuquerque, NM USA
colbaugh@comcast.net

Kristin Glass
New Mexico Institute of Mining and Technology
Socorro, NM USA
kglass@icasa.nmt.edu

***Abstract*—There are substantial potential benefits to considering *predictability* when designing defenses against adaptive adversaries, including increasing the ability of defense systems to predict new attacker behavior and reducing the capacity of adversaries to anticipate defensive actions. This paper adopts such a perspective, leveraging the coevolutionary relationship between attackers and defenders to derive methods for predicting and countering attacks and for limiting the extent to which adversaries can learn about defense strategies. The proposed approach combines game theory with machine learning to model adversary adaptation in the learner's feature space, thereby producing classes of predictive and "moving target" defenses which are scientifically-grounded and applicable to problems of real-world scale and complexity. Case studies with large cyber security datasets demonstrate that the proposed algorithms outperform gold-standard techniques, offering effective and robust defense against evolving adversaries.**

***Keywords*—**-predictive defense, moving target defense, game theory, machine learning, adaptive adversaries, cyber security.

## I. Introduction

Adaptive adversaries are a principal concern in many security domains, including cyber defense, border security, counterterrorism, and crime prevention [e.g. 1-3]. Consequently, there is great interest in developing defenses which maintain their effectiveness despite evolving adversary strategies and tactics. A potentially powerful approach to pursuing such goals is to explicitly consider system *predictability*, for instance in order to design defenses which are able to anticipate adversary behavior and/or decrease their own predictability. Studies that employ predictability assessment in a security context include [4,5].

The coevolving "arms race" between Spammers and Spam filters provides an illustrative example of the phenomenon of interest [e.g. 6,7]. Spam filter designers would like to produce filters that work well against both present and future Spam, and one way to accomplish this goal is to develop techniques for predicting the way Spammers will adapt to currently-deployed filters and to account for these expected adaptations during the filter design process. Spammers, on the other hand, are motivated to "reverse-engineer" existing Spam filters as quickly as possible, so they can generate Spam which circumvents these filters. Spam filter developers are therefore interested in both sides of the predictability question: they wish to construct filters that can predict (and defeat) new Spammer techniques while remaining unpredictable themselves. Many other security problems involve adaptive adversaries and coevolutionary dynamics, and we propose that valuable insights can be obtained by examining these dynamics through the lens of predictability; Spam is merely a simple, familiar example of such systems.

Because predictability-based defense design includes strategic considerations, it is natural to approach this design problem as a game [8], in which defense attempts to predict and counter adversary behaviors while reducing its own predictability. Unfortunately, previous attempts to apply game-theoretic methods to adversary defense [e.g. 9-15] have encountered a number of challenges, and we mention two that have been especially daunting. First, the set of possible attacker actions is typically very large in real-world settings, and because the complexity of most game models increases exponentially with the number of actions available to the players, this has often made these models intractable in practice. And second, it has proved difficult to derive models that capture evolving attacker behavior in any but the most idealized situations.

In this paper we overcome these challenges by developing our game-based models for attack-defend interaction within a machine learning (ML) framework [16], enabling the design of robust defenses for practical applications. We formulate the defense task as one of behavior classification, in which innocent and malicious activities are to be distinguished, and assume only limited information is available regarding prior attacker behavior or attack attributes. The defense's classifiers model attacker actions in ML *feature space*, that is, in the space of variables the ML algorithms use for learning and decision-making. Formulating attack prediction/defense synthesis in this "compressed" and abstract space enables derivation of algorithms that can be applied to practical, large-scale problems.

The first of the proposed defense systems explicitly attempts to predict and counter adversary adaptation as a means of providing effective defense against both current and future attacks. A key step in the approach is modeling the way attackers *adapt* their behaviors rather than modeling the behaviors themselves. Crucially, the proposed approach seeks to design optimal defenses for evolving attacks, rather than to predict new attacks perfectly, and therefore enjoys robust performance in the presence of (inevitable) prediction errors. To permit the performance of this predictive defense method to be evaluated, we have assembled for this investigation a large collection of Spam and non-Spam emails reflecting the evolution of Spammer tactics over an eight year period. A case study with this dataset demonstrates that the proposed defense significantly outperforms a gold-standard Spam filter.

An important consideration when applying classifier-based defense techniques, even predictive ones, is the extent to which adversaries can reverse-engineer the learning algorithm and use this knowledge to circumvent the defense. The goal of the second proposed defense is thus to reduce defense system predictability and increase the difficulty of the adversary's reverse-engineering task. We adopt a "moving target" (MT) perspective, in which the defense presents a dynamic posture to the adversaries as a way of increasing the adversaries' uncertainty concerning defense operation [17]. By leveraging recent advances in the theory of repeated, incomplete information games [18,19], we derive a simple MT defense procedure which can be shown to be optimal for an important class of adversarial dynamics; interestingly, the optimal MT schedule can be specified independently of the details of the adversaries' strategies. The efficacy of the proposed MT defense is evaluated via case studies with the set of Spam and non-Spam emails mentioned above and also with a well-known publicly-available network intrusion dataset. These tests reveal that the MT defense substantially outperforms well-tuned static classifiers against adaptive adversaries.

## II. Predictive Defense

### A. Problem Formulation

There are significant potential benefits to developing *predictive* methods of defending against adaptive adversaries, in which opponents' evolving strategies are anticipated and these insights are employed to counter novel attacks. This section considers the following concrete instantiation of the predictive defense problem: given some history of attacker actions, design a defense system which performs well against both current and future attacks. It is reasonable to expect that concepts and techniques from game theory might be helpful in understanding adversary adaptation, and indeed such approaches have been explored in a variety of domains [e.g. 9-15]. However, as indicated in the Introduction, these investigations have encountered scalability and complexity challenges which have limited their practical utility. In this section we address these challenges by deriving our game-based model within an ML framework, enabling effective defense in realistic settings. (See [20] for a general discussion of the value of combining behavioral modeling with data mining algorithms for discovery and prediction applications.)

We approach the task of countering adversarial behavior as an ML classification problem, in which the objective is to distinguish innocent and malicious activity. Each instance of activity is represented as a feature vector $x \in \Re^{|F|}$, where entry $x_i$ of x is the value of feature i for this instance and F is the set of instance features. In what follows, F is a set of "reduced" features, obtained by projecting measured feature vectors into a lower-dimensional space. While feature reduction is standard practice in ML [16], we show below that *aggressive* reduction allows us to efficiently manage the complexity of our game models. Behavior instances x belong to one of two classes: positive/malicious and negative/innocent (generalizing to more than two behavior classes is straightforward [16]). The goal is to learn a vector $w \in \Re^{|F|}$ such that classifier orient = sign($w^T x$) accurately estimates the class of behavior x, returning +1 (−1) for malicious (innocent) activity.

As indicated above, it is useful to assess the predictability of a phenomenon before attempting to predict its evolution; for example, such an analysis permits identification of measurables which possess predictive power [21]. There has been limited theoretical work assessing predictability of adversarial dynamics, but existing studies suggest attack-defend coevolution often generates predictable dynamics. For instance, although [22] finds that certain player strategies lead to chaos in a simple repeated game, [20] shows that large sets of player strategies and repeated games exhibit predictable adversarial dynamics. Here we supplement this theoretical work by conducting an empirical investigation of predictability, and select as our case study a cyber security problem – Spam filtering – which possesses attributes that are representative of many adversarial domains.

To conduct this investigation, we first obtained a large collection of emails from various publicly-available sources for the period 1999-2006, and added to this corpus a set of Spam emails acquired from B. Guenter's Spam trap for the same time period. Following standard practice, each email is modeled as a "bag of words" feature vector $x \in \Re^{|F|}$, where the entries of x are the frequencies with which the words in vocabulary F appear in the message. The resulting dataset consists of ~128,000 emails composed of more than 250,000 features. We extracted from this collection of Spam and non-Spam emails the set of messages sent during the 30 month period between January 2001 and July 2003 (email in other periods exhibit very similar evolutionary dynamics). Finally, the dimension of the email feature space was reduced via a singular value decomposition (SVD) analysis [16], yielding a reduction in feature space dimension of four orders of magnitude (from ~250K to 20).

We wish to examine, in a simple but meaningful way, the predictability of Spam adaptation, and propose two intuitively reasonable criteria with which to empirically evaluate predictability: *sensibility* and *regularity* (a comprehensive theoretical framework for defining and assessing predictability is given in [21]). More specifically, and in the context of Spam, it would be *sensible* for Spammers to adapt their messages over time in such a way that Spam feature vectors $x_S$ come to resemble the feature vectors $x_{NS}$ of legitimate emails, and *regularity* in this adaptation might imply that the values of the individual elements of $x_S$ approach those of $x_{NS}$ in a fairly monotonic way.

To permit convenient examination of the evolution of feature vectors $x_S$ and $x_{NS}$ during the 30 month period under study, the emails were first binned by quarter. Next, the average values for each of the 20 (reduced) features was computed for all the Spam emails and all the non-Spam emails (separately) for each quarter. Figure 1 illustrates the feature space dynamics of Spam and non-Spam messages for one representative coordinate (F1) of this reduced feature space. It can be seen in the plot that the value of feature F1 for Spam approaches the value of this feature for non-Spam, and this increasing similarity is a consequence of changes in the composition of Spam messages (the value of F1 for non-Spam emails is essentially constant). The dynamics of the other feature values are analogous.

Observe that the Spam dynamics illustrated in Figure 1 reflect *sensible* adaptation on the part of Spammers: the features of Spam email messages evolve to appear more like those of non-Spam email, making Spam more difficult to detect. Additionally, this evolution is *regular*, with feature values for Spam approaching those for non-Spam in a nearly-monotonic fashion. Thus this empirical analysis indicates that coevolving Spammer-Spam filter dynamics possesses some degree of predictability, and that the features employed in Spam analysis may have predictive power; this result is in general agreement with the conclusions of the theoretical predictability analysis reported in [20]. Moreover, because many of the characteristics of Spam-Spam defense coevolution are shared by other adversarial systems, this result suggests these other systems may have exploitable levels of predictability as well.

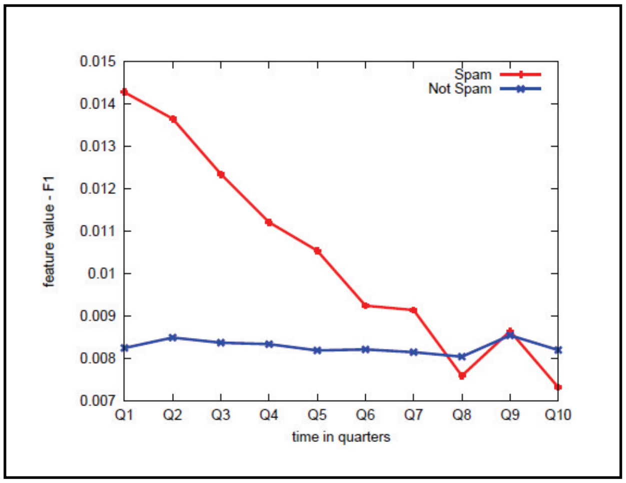


**Figure 1.** Spam/non-Spam evolution in feature space. The plot depicts evolution of feature F1 for Spam (red) and non-Spam (blue) over time (horizontal axis).

### *B. Predictive Defense Algorithm*

The proposed approach to designing a predictive defense system which works well against both current and future attacks is to combine ML with a simple game-based model for adversary behavior. In order to apply game-theoretic methods, it is necessary to overcome the complexity and model-realism challenges mentioned above. We address problem complexity by modeling adversary actions directly in an aggressively-reduced ML feature space, so that the (effective) space of possible adversary actions which must be considered is dramatically decreased. The difficulty of deriving realistic representations for attacker behavior is overcome by recognizing that the actions of attackers can be modeled as attempts to *transform* data (i.e., feature vectors x) in such a way that malicious and innocent activities are indistinguishable. (This is in contrast to trying to model the attack instances "from scratch"). It is possible to model attacker actions as transformations of data because, within an ML problem formulation, historical attack data are available in the form of training instances.

We model adversarial coevolution as a sequential game, in which the attacker and defender iteratively optimize the following objective function:

$$\min_{w} \max_{a} \left[ -\alpha \|a\|^3 + \beta \|w\|^3 + \sum_{i} \text{loss}\left(y_i, w^T (x_i + a)\right) \right] \qquad (1)$$

In (1), the loss function represents the misclassification rate for the defense system, where $\{y_i, x_i\}^n_{i=1}$ denotes pairs of "nominal" activity instances $x_i$ and labels $y_i$, and vector w parameterizes the defense (recall that the defense attempts to distinguish malicious and innocent activity using the classifier orient = sign($w^T$x)). The attacker attempts to circumvent the defense by transforming the data through vector $a \in \Re^{|F|}$, and the defender's goal is to counter this attack by appropriately specifying classifier vector $w \in \Re^{|F|}$. The terms $-\alpha\|a\|^3$ and $\beta\|w\|^3$ define "regularizations" imposed on attacker and defender actions, respectively, as discussed below.

Note that (1) models the attacker as acting to increase the misclassification rate with vector a, subject to the need to limit the magnitude of this vector (large a is penalized via the term $-\alpha\|a\|^3$). This model thus captures in a simple way the fact that the actions of the attacker are in reality always constrained by the goals of the attack. For instance, in the case of Spam email attacks, the Spammer tries to manipulate message x in such a way that it "looks like" legitimate email and evades the Spam filter w. However, transformed message x+a must still communicate the desired information to the recipient or the attacker's goal will not be realized, and so the transformation vector a cannot be chosen arbitrarily.

The defender attempts to reduce the misclassification rate with an optimal choice for vector w, and avoids "over-fitting" through regularization with the $\beta\|w\|^3$ term [16]. Notice that the formulation (1) permits the attacker's goal to be modeled as counter to, but not exactly the opposite of, the defender's goal, and this is consistent with many real-world settings. Returning to the Spam example, the Spammer's objective of delivering messages which induce profitable user responses is not the inverse of an email service provider's goal of achieving high Spam recognition with a very low false-positive rate.

The preceding development can be summarized by stating the following predictive defense (PD) algorithm:

**Algorithm PD**

1. Collect historical data $\{y_i, x_i\}^n_{i=1}$ which reflects past behavior of the attacker as well as past legitimate behavior.
2. Optimize objective function (1) to obtain the predicted actions a* of the attacker and the optimal defense w* to counter this attack.
3. Estimate the status of any new activity x as either malicious (+1) or innocent (−1) via orient = sign($x^T$w*).

Observe that Step 2 of this algorithm can be interpreted as first predicting the attacker strategy through computation of attack vector a*, and then learning an appropriate countermeasure w* by applying ML to the "transformed" data $\{y_i, x_i+a^*\}^n_{i=1}$.

### *C. Algorithm Evaluation*

This case study examines the performance of Algorithm PD for the Spam filtering problem. We use the Spam/non-Spam email dataset introduced above, consisting of ~128,000 messages that were sent during the period 1999-2006. The study compares the effectiveness of Algorithm PD, implemented as a Spam filter, with that of a well-tuned naïve Bayes (NB) Spam filter [5]. Because NB filters are widely used and work very well in Spam applications, this filter is referred to as the gold-standard algorithm. We extract from our dataset the 1000 oldest legitimate emails and 1000 oldest Spam messages for use in training both Algorithm PD and the gold-standard algorithm. The email messages sent during the four year period immediately following the date of the last training email are used as test data. More specifically, these emails are binned by quarter and then randomly sub-sampled to create balanced datasets of Spam and legitimate emails for each of the 16 quarters in the test period.

Recall that Algorithm PD employs aggressive feature space dimension reduction to manage the complexity of the game-based modeling process. This dimension reduction is accomplished here through SVD analysis, which reduces the dimension |F| of feature vectors from ~250K to 20) [16]. (The orthogonal basis used for this reduction is derived by performing SVD analysis using the 1000 non-Spam and 1000 Spam training emails.) Note that good classification accuracy can be obtained with a wide range of (reduced) feature space dimensions. For example, a filtering accuracy of ~97% is achieved with the training data when using an NB classifier implemented with feature dimension ranging from |F|=100,000 to |F|=5.

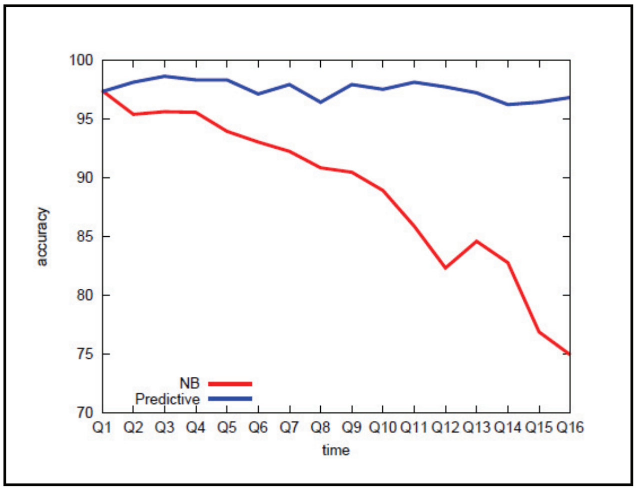


**Figure 2.** Results for predictive defense Spam filtering case study. The plot shows how filter accuracy (vertical axis) varies with time (horizontal axis) for the gold-standard NB filter (red) and Algorithm PD filter (blue).

The gold-standard strategy is applied as described in [5]. Algorithm PD is implemented with parameter values $\alpha = 0.001$ and $\beta = 0.1$, and with a sum-of-squares loss function. To evaluate the utility of the defenses against evolving adversaries, we train Algorithm PD and the gold-standard algorithm *once*, using the 1000 non-Spam/1000 Spam dataset, and then apply the filters without retraining to the four years of emails that follow these 2000 emails.

Sample results from this study are depicted in Figure 2. Each data point in the plots represents the average accuracy over ten trials (two-fold cross-validation). It can be seen that the filter based upon Algorithm PD significantly outperforms the gold-standard method: the predictive defense experiences almost no degradation in accuracy over the four years of the study, while the gold-standard method suffers a substantial drop in accuracy during this period. These results suggest that combining ML with simple game-based models offers an effective means of defending against evolving adversaries.

## III. Moving Target Defense

### *A. Problem Formulation*

A defining characteristic of classification-based defense is the fact that adversaries continually attempt to reverse-engineer the classifier and use this knowledge to make informed adjustments to their behavior and circumvent the defense. One way to increase the difficulty of the adversary's reverse-engineering task is to employ moving target (MT) ideas, in which the defense adopts a time-varying posture in order to increase adversary uncertainty concerning defense operation [17]. In this section we derive an MT defense procedure which minimizes the predictability of defensive actions from the perspective of the attacker.

We investigate MT defense within the framework provided by two-player repeated games with incomplete information [18,19]. In these games one player, the *informed* player, has access to information that is unavailable to the other, *uninformed*, player. The informed player must weigh the relative benefits of exploiting her private information to achieve short-term advantage against the possibility that this exploitation may reveal information which results in the sacrifice of future gains. Because repeated incomplete information games explicitly account for the payoff-predictability tradeoff, they afford a convenient setting for deriving and comparing MT strategies.

Consider the following defense problem. Suppose the task of countering adversarial behavior is formulated as one of ML classification, in which the objective is to distinguish innocent and malicious activity. Each instance of activity is represented as a feature vector $x \in \Re^{|F|}$, where F is the set of ML features. Behavior instances x belong to one of two classes, positive/malicious and negative/innocent, and the goal is to learn a vector $w \in \Re^{|F|}$ such that classifier class = sign($w^T x$) accurately estimates the class of behavior x.

A plausible way to reduce the degree to which adversaries can predict, and then adapt to and evade, the actions of a classifier is to introduce randomness into the way the ML features F are selected and used. One simple way to accomplish this is delineated in the following three steps: 1.) divide the original feature set F into K randomly-selected, possibly overlapping subsets $\{F_1, \ldots, F_K\}$, where $|F_i| = m \ \forall\ i$; 2.) train one classifier for each feature subset $F_i$, yielding a collection of K classifiers $\{w_1, \ldots, w_K\}$; 3.) during operation, alternate between the classifiers $w_i$ according to some randomized scheduling policy. In order to implement this MT defense, it is necessary to define a procedure for selecting which classifier is to be "active" at each time period. Thus the MT defense problem of interest can be stated: given a collection of classifiers $W=\{w_1, \ldots, w_K\}$, specify a policy for switching among classifiers which minimizes defense predictability (from the point of view of the attacker).

### B. *Moving Target Scheduling Policy*

A classifier schedule which minimizes defense predictability is sketched in the following theorem. Perhaps surprisingly, the optimal schedule is very simple to implement.

**Theorem MT:** Suppose we are given a collection of K classifiers $W = \{w_1, \ldots, w_K\}$ associated with randomly-selected feature subsets $\{F_1, \ldots, F_K\}$, an ecology of adversaries that wish to reverse-engineer the defense, and a sequence of times $t_1, t_2, \ldots$ at which it is permissible to switch classifiers. Under mild assumptions regarding the accuracy of the classifiers W prior to adversary reverse-engineering and the effectiveness of the reverse-engineering methods, defense performance is optimized if, at each time $t_i$, the active classifier $w_a$ is selected uniformly at random from the set W.

**Proof:** The proof is given in [23]. ■

We now provide a concise, intuitively-accessible summary of the proof of Theorem MT. Additionally, we describe empirical tests of the theorem's conclusions in Section IIIC below. Readers interested in the technical details of the proof are referred to the report [23]. We model the interaction between an MT defense and an ecology of adversaries as a *hidden mode hybrid dynamical system* (HM-HDS) (see, for instance, [19] for background on this class of dynamical systems). More precisely, the MT defense model is

$$\Sigma_{\text{HM-HDS}} = \{C(w,a), W, P(w,a)\} \qquad (2)$$

where

- the *continuous system* C(w,a) evolves according to sequential attack-defend game dynamics (such as (1));
- the *discrete system* {W,P(w,a)} evolves as a Markov chain with state set W (the set of candidate classifiers) and state transition probability matrix P(w,a); note that, in general, state transition probabilities may depend upon the continuous system state variables (w,a);
- the *hidden mode* is the discrete system state, that is, the currently active classifier $w_a \in W$.

A schematic of this HM-HDS model is depicted in Figure 3.

The dynamics of the HM-HDS (2) evolve as follows. The discrete system specifies the currently active classifier $w_a$, and this information is communicated to the defender (but not the attacker) in the continuous system game. The attacker attempts to infer which classifier is active by observing defense actions, and computes attack vector a based on this estimate. The discrete system has access to continuous system state (w,a) and may use this information when choosing the next active classifier.

We interpret these dynamics as a repeated incomplete information game, in which the discrete system is the informed player and the attacker dynamics is the uninformed player [18]. (This formulation, although less intuitive than the two-player game model adopted in Section II, facilitates analysis of MT dynamics.) The payoff to the discrete system is defined to be the negative of the misclassification rate, so that maximizing this payoff is equivalent to maximizing the performance of the defense.

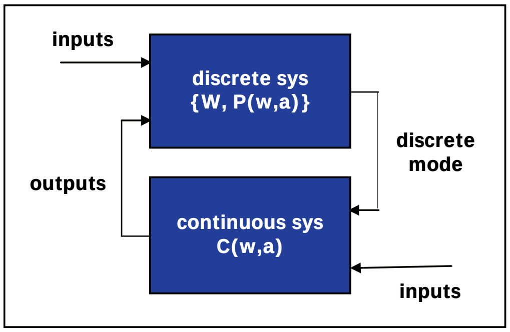


**Figure 3.** Schematic of basic HM-HDS feedback structure. The discrete and continuous systems in this framework model the selection of "active" classifier $w_a$ and the resulting attack-defend dynamics, respectively.

Now suppose: 1.) each classifier $w_i \in W$ is effective against nominal, "pre-reverse-engineering" attacks (they need not be equally effective), and 2.) the attackers collectively have good reverse-engineering capabilities (i.e., reverse-engineering produces a substantial drop in classifier accuracy for each $w_i \in W$); these conditions are defined more quantitatively in [23]. Under these assumptions, $\Sigma_{\text{HM-HDS}}$ (2) belongs to a class of HM-HDS which is studied in [19]. In that paper, the control of such HM-HDS is formulated as an incomplete information game between a "controller" (the uninformed player) and a "disturbance" (the informed player), where the actions of the disturbance can reveal to the controller exploitable information about the current value of the discrete mode. It is shown in [19] that, in this setting, the best strategy for the disturbance is to maximize the controller's uncertainty regarding the (hidden) discrete mode. This result in turn implies that, in the case of MT defense system (2), the optimal scheduling policy for discrete system {W, P(w,a)} is to select the active classifier $w_a$ uniformly at random from the set W at each time $t_i$.

Observe that the optimal choice of a new $w_a$ does not depend upon the currently active classifier or the continuous state variables (w,a), basically because any such dependence has the potential to be exploited by the attacker. Additionally, and perhaps counterintuitively, each of the classifiers $w_i$ has an equal probability of being selected to be active, even though some may be more accurate that others. Roughly, if classifier w* is implemented with greater frequency than the others, say because it is especially accurate, the attackers will have increased opportunity to successfully reverse-engineer it, rendering w* less effective than the others in the long run.

### C. *Algorithm Evaluation: Spam*

In this section we evaluate the effectiveness of the MT defense strategy summarized in Theorem MT by employing the Spam filtering data and task introduced in Section II. To facilitate convenient comparison with gold-standard defense systems and to reduce complications in the assessment, a few simplifications are made:

- standard NB Spam filters are used for the classifiers $\{w_1, \ldots, w_K\}$ (rather than using, say, the predictive filters generated by solving (1));
- only K=2 classifiers/feature subsets are used;
- attack vector a is computed in an optimal manner via (1), so that the adversary possesses strong reverse-engineering capabilities.

To enable the efficacy of the proposed MT defense to be quantified, its performance is compared to that of a well-tuned static NB filter trained using the full set of (reduced-dimension) features F. We examine a range of attack "strengths" by varying the parameter $\alpha$ in the optimization (1) (recall that the term $-\alpha||a||^3$ governs the magnitude of attack vector a). Attacks are normalized by assigning an attack strength of AS=1 to attacks with magnitude $||a||$ equal to the largest attack observed in the (real-world) Spam dataset.

We apply the static NB filter and the optimal two-mode (K=2) MT filter to the 2000 email training dataset described in Section IIC. Additionally, to allow the results of Theorem MT to be tested, we implement a suboptimal MT filter obtained by favoring the more accurate of the two classifiers in the random scheduling process; specifically, the more accurate of the two filters is selected to be active with 2/3 probability (with the less accurate filter then being selected 1/3 of the time). Feature set F is taken to be the collection of 20 features with largest singular values (see Section IIC), and feature subsets $F_1$ and $F_2$ are constructed by randomly sampling F (with replacement) until each subset contains 10 features. The filters are "attacked" by solving (1) for the optimal attack a* and then transforming Spam instances x according to the formula x+a*. To allow exploration of a range of attack strengths, (1) is solved for different values of $\alpha$, yielding the following AS values: AS=0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5 (thus attacks vary in strength from 'no attack' to attacks with magnitude 1.5 times larger than any seen in the Spam dataset).

Sample results are displayed in Figure 4. Each data point in the plots represents the average accuracy over ten trials (two-fold cross-validation). It can be seen that the filter based upon Theorem MT (red curve) significantly outperforms the static NB filter (magenta curve). For instance, MT defense achieves a classification accuracy of ~90% when subjected to attacks of strength AS=1, compared with the ~65% accuracy obtained with the static filter. Under attacks of magnitude AS=1.5 the optimal MT defense provides an accuracy of ~80%, while the static filter is only slightly more effective than random guessing in this case (accuracy ≈ 54%).

Moreover, this empirical study offers support for the conclusions of Theorem MT. As can be seen from Figure 4, the filter which schedules the more accurate classifier with greater probability (blue curve) does not perform as well as the optimal (according to Theorem MT) MT filter, particularly when the filters are subjected to fairly strong attacks corresponding to effective adversary reverse-engineering. These results suggest that the proposed MT defense is capable of substantially increasing the difficulty of reverse-engineering tasks, even for highly effective (e.g., optimal) attackers.

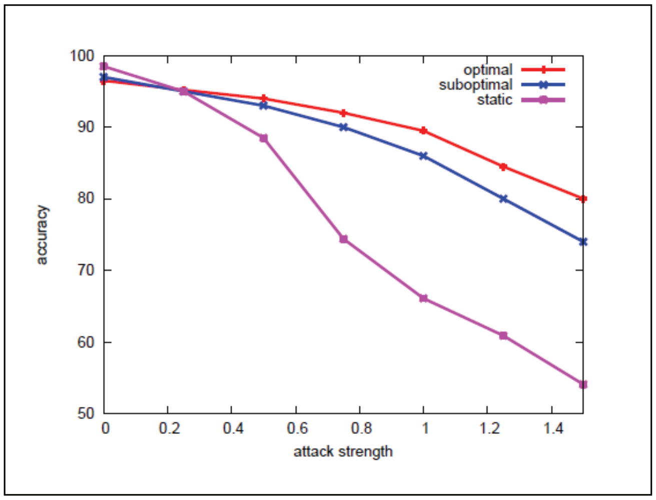


**Figure 4.** Results for moving target defense Spam filtering case study. The plot shows how filter accuracy (vertical axis) varies with attack strength (horizontal axis) for the optimally scheduled MT filter (red), a suboptimally scheduled MT filter (blue), and the static NB filter (magenta).

### D. *Algorithm Evaluation: Network Intrusion*

We now examine the performance of the MT defense strategy summarized in Theorem MT for the problem of distinguishing innocent and malicious computer network activity. The empirical data used for this case study is the KDD Cup 99 dataset, a publicly-available collection of network data consisting of both normal activities and attacks of various kinds [24]. For this study we randomly selected 1000 Normal connections (N) and 1000 denial-of-service attacks (DoS) to serve as our test data.

To enable the efficacy of the proposed MT defense to be quantified, its performance is compared to that of a well-tuned static NB classifier [5]. This NB classifier uses the full set of 30 "continuous" features adopted in previous studies (see, e.g., [5] for a discussion). The optimal two-mode (K=2) MT classifier employs feature subsets $F_1$ and $F_2$ constructed by randomly sampling F (with replacement) until each subset contains 15 features. The classifiers are attacked by solving (1) for the optimal attack a* and then transforming DoS network activity instances x according to the formula x+a*. As in the preceding case study, we obtain a range of attack strengths by solving (1) for different values of $\alpha$ (recall $-\alpha||a||^3$ governs the magnitude of attack vector a).

Sample results are displayed in Figure 5. Each data point in the plots represents the average accuracy over ten trials (two-fold cross-validation). It can be seen that the classifier based upon Theorem MT (blue curve) significantly outperforms the static NB classifier (red curve). For instance, the accuracy of the MT defense system never goes below 90%, even when subjected to large attacks, while the accuracy of the static defense quickly falls to 50% as attack strength is increased (this corresponds to random guessing, as the dataset is balanced). Note that this case study illustrates the ease with which the proposed approach can be implemented in different adversarial settings.

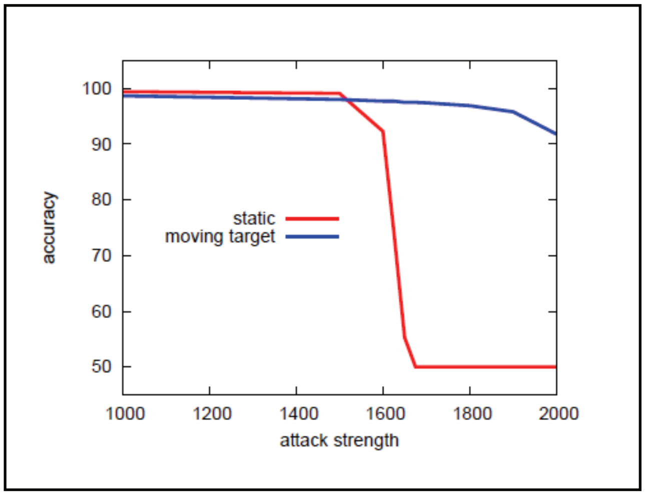


**Figure 5.** Results for moving target defense network intrusion case study. The plot shows how classifier accuracy (vertical axis) varies with attack strength (horizontal axis) for the optimally scheduled MT defense (blue) and static NB defense (red).

## ACKNOWLEDGEMENTS

This work was supported by the Laboratory Directed Research and Development Program at Sandia National Laboratories. We thank Chip Willard and Curtis Johnson for numerous helpful discussions on aspects of this research.

## REFERENCES

[1] *Proc. 2010 IEEE International Conference on Intelligence and Security Informatics*, Vancouver, BC Canada, May 2010.

[2] *Proc. 2011 IEEE International Conference on Intelligence and Security Informatics*, Beijing, China, July 2011.

[3] *Proc. 2012 IEEE International Conference on Intelligence and Security Informatics*, Washington, DC USA, June 2012.

[4] Colbaugh, R., "Does coevolution in malware adaptation enable predictive defense?", *IFA Workshop Series: Exploring Malware Adaptation Patterns*, San Francisco, CA, May 2010.

[5] Colbaugh, R. and K. Glass, "Proactive defense for evolving cyber threats", *Proc. 2011 IEEE ISI*, Beijing, China, July 2011.

[6] Cormack, G., "Email Spam filtering: A systematic review", *Foundations and Trends in Information Retrieval*, Vol. 1, pp. 335-455, 2008.

[7] Guzella, T. and W. Caminhas, "A review of machine learning approaches to Spam filtering", *Expert Systems with Application*, Vol. 36, pp. 10206-10222, 2009.

[8] Peters, H., *Game Theory*, Springer, Berlin, 2008.

[9] Dalvi, N. et al., "Adversarial classification", *Proc. ACM KDD '04*, Seattle, WA, August 2004.

[10] Roy, S. et al., "A survey of game theory as applied to network security", *Proc. HICSS 2010,* Honolulu, HI, January 2010.

[11] Williams, E., *Surveillance and Interdiction Models: A Game Theoretic Approach to Defend Against VBIED*, Thesis, Naval Postgraduate School, June 2010.

[12] Parameswaran, M., H. Rui, and S. Sayin, "A game theoretic model and empirical analysis of Spammer strategies", *Proc. CEAS 2010,* Redmond, WA, July 2010.

[13] Gkonis, K. and H. Psaraftis, "Container transportation as an interdependent security problem", *J. Transportation Security*, Vol. 3, pp. 197-211, 2010.

[14] Pita, J. et al., "GUARDS: Game theoretic security allocation on a national scale", *Proc. AAMAS '11*, Taipei, Taiwan, May 2011.

[15] Manshaei, M. et al., "Game theory meets network security and privacy", *ACM Computing Surveys*, December 2011.

[16] Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Second Edition, Springer, New York, 2009.

[17] *Trustworthy Cyberspace: Strategic Plan for the Federal Cybersecurity Research and Development Program*, December 2011.

[18] Sandholme, T., "State of solving large incomplete information games, and application to poker", *AI Magazine*, pp. 13-32, 2010.

[19] Verma, R. and D. Del Vecchio, "Safety control of hidden mode hybrid systems", *IEEE Trans. Automatic Control*, Vol. 57, pp. 62-77, 2012.

[20] Colbaugh, R., "Arctic ice, George Clooney, lipstick on a pig, and insomniac fruit flies: Combining kd and m&s for predictive analysis", *Proc. ACM KDD '11*, San Diego, CA, August 2011.

[21] Colbaugh, R. and K. Glass, "Predictive analysis for social processes I: Multi-scale hybrid system modeling, and II: Predictability and warning analysis", *Proc. 2009 IEEE MSC*, Saint Petersburg, Russia, July 2009.

[22] Sato, Y., E. Akiyama, and J.D. Farmer, "Chaos in learning a simple two-person game", *Proc. National Academy of Sciences USA*, Vol. 99, pp. 4748-4751, 2002.

[23] Colbaugh, R. and K. Glass, "Predictive dynamic defense against adaptive adversaries", Sandia National Laboratories Technical Report, April 2012.

[24] http://kdd.ics.uci.edu/databases/kddcup99/; accessed Dec. 2010.

# Leveraging Sociological Models for Prediction I: Inferring Adversarial Relationships

Richard Colbaugh
Sandia National Laboratories
Albuquerque, NM USA
colbaugh@comcast.net

Kristin Glass
New Mexico Institute of Mining and Technology
Socorro, NM USA
kglass@icasa.nmt.edu

***Abstract*—There is considerable interest in developing techniques for predicting human behavior, for instance to enable emerging contentious situations to be anticipated or permit the nature of ongoing but "hidden" activities to be inferred. A promising approach to this problem is to collect appropriate empirical data and then apply machine learning methods to the data to generate the predictions. This two-part paper shows that the performance of such learning algorithms often can be improved substantially by leveraging sociological models in their development and implementation. In particular, we demonstrate that sociologically-grounded learning algorithms outperform gold-standard methods in two important and challenging tasks: 1.) inferring the (unobserved) nature of relationships in adversarial social networks, and 2.) predicting whether nascent social diffusion events will "go viral". Significantly, the new algorithms perform well even when there is limited data available for their training and execution.**

***Keywords*—**predictive analysis, sociological models, social networks, machine learning.

## I. Introduction

There is great interest in developing techniques for accurately predicting human behavior. For example, forecasting the eventual outcomes of social processes is a central concern in domains ranging from popular culture to public policy to national security [1]. The task of inferring the existence and nature of activities which are presently underway but not directly observable, sometimes referred to as "predicting the present" [2], is also of crucial importance in many applications. A promising approach to obtaining such predictions is to identify and collect empirical data which appropriately characterize the phenomenon of interest and then to analyze these data using machine learning (ML) methods [3]. Roughly speaking, ML algorithms automatically "learn" relationships between observed variables from examples presented in the form of training data; the learned relationships are then used to generate predictions in new situations. ML's capacity to learn from examples, scale to large datasets, and adapt to new or changing conditions make this an attractive approach to predictive analysis.

The work reported in [4-12] illustrates some of the ways ML can be used for forecasting, and in particular how these techniques can be applied to online (Web) data in order to predict the outcomes of a broad range of social processes (e.g., social movements, political elections and protests, and markets of various kinds). Alternatively, the papers [13-19] derive ML techniques for predicting the present, for instance enabling the existence of hidden links in social networks to be inferred, the sentiment of informal communications to be estimated, and the spread of various health-related phenomena to be remotely monitored and assessed.

Existing ML methods, although very useful, face at least two key challenges. First, the prediction accuracy obtainable with even state-of-the-art algorithms is sometimes insufficient for the task at hand, such as when the predictions are to be used to inform high-consequence decisions (e.g., pertaining to national security or human health). Second, applying ML techniques typically requires that significant quantities of data be collected and "labeled". For example, deriving an ML scheme for estimating sentiment polarity of blog posts usually involves collecting, processing, and manually labeling hundreds of example posts expressing positive and negative sentiment [14]. Employing ML for forecasting ordinarily entails assembling extensive time series traces, implying that such methods may not be responsive enough to generate useful predictions about rapidly emerging events [12]. Additionally, realizing good performance with standard ML usually necessitates frequent retraining to permit algorithms to adapt to evolving conditions, which limits usefulness in many domains (e.g., in adversarial settings in which opponents adapt their behaviors expressly to defeat learning algorithms [20]).

This two-part paper proposes that the challenges of predicting human behavior using ML often can be overcome by leveraging sociological models in the development and implementation of the learning algorithms. This proposal is motivated by our recent research which shows that including sociologically-meaningful measures of network dynamics as features in ML algorithms permits predictions regarding social dynamics that are substantially more accurate than those based on standard features [21]. The present two-part paper initiates a more systematic exploration of the utility of combining ML with sociological models for social prediction. In Part One, we consider the problem of predicting the "signs" of relationships in social networks, where positive and negative edges reflect friendly and antagonistic social ties, respectively, and derive a novel ML algorithm for edge-sign prediction which leverages structural balance theory [22-24]. The proposed algorithm outperforms a "gold-standard" method in empirical tests with two large-scale online social networks, with the boost in prediction accuracy being especially significant in situations where only limited training date are available. Interestingly, the inferred edge-signs are also shown to be useful when predicting the way adversarial networks will fracture under stress.

Part Two of the paper [25] examines the problem of forecasting the ultimate reach of "complex contagion" events [26]. Predictability assessment of such contagions indicates that the metrics which should be predictive of a contagion's reach are subtle measures of the network dynamics associated with very early diffusion activity. These results are used to derive an ML algorithm for predicting which complex contagions will eventually "go viral" and which won't, and it is demonstrated that the algorithm outperforms standard methods in an empirical investigation of online meme propagation [27]. Significantly, the new algorithm performs well even when only limited time series data are available for analysis, permitting reliable prediction early in the contagion lifecycle. It is also shown that the proposed algorithm enables effective early warning analysis for an important class of cyber threats.

## II. Predicting Link-Signs

### A. Problem Formulation

Social networks may contain both positive and negative relationships – people form ties of friendship and support but also of animosity or disapproval. These two types of social ties can be modeled by placing signs on the links or edges of the social network, with +1 and –1 reflecting friendly and antagonistic relationships, respectively. We wish to study the problem of predicting the signs of certain edges of interest by observing the signs and connectivity patterns of the neighboring edges. More specifically, for a directed social network $G_s$ = (V, E) with signed edges, where V and E are the vertex and edge sets, we consider the following edge-sign prediction problem: given an edge $(u,v)\in E$ that is of interest but for which the edge-sign is "hidden", infer the sign of (u,v) using information contained in the remainder of the network.

It is natural to suspect that *structural balance theory* (SBT) may be useful for edge-sign prediction. Briefly, SBT posits that if $w\in V$ forms a *triad* (i.e., edge triangle) with edge (u,v), then the sign of (u,v) should be such that the resulting signed triad possessing an odd number of positive edges; this encodes the common principle that "the friend of my friend is my friend", "the enemy of my friend is my enemy", and so on [22,23]. Thus SBT suggests that knowledge of the signs of the edges connecting (u,v) to its neighbors may be useful in predicting the sign of (u,v).

### B. Prediction Algorithm

We approach the task of predicting the sign of a given edge (u, v) in the social network $G_s$ as an ML classification problem. The first step is to define, for a given edge, a collection of features which may be predictive of the sign of that edge. To allow a comparison with the (gold-standard) prediction method given in [24], we adopt the same two sets of features used in that study. For a given edge (u,v), the first set of features defined in [24] characterize the various triads to which (u,v) belongs. Because triads are directed and signed, there are sixteen distinct types (e.g., the triad composed of positive edge (u,w) and negative edge (w,v), together with (u,v), is one type). Thus the first sixteen features for edge (u,v) are the counts of each of the various triad types to which (u,v) belongs. Including these features is directly motivated by SBT. For example, if (u,v) belongs to many triads with one positive and one negative edge, it may be likely that the sign of (u,v) is negative, since then these triads would possess an odd number of positive edges and therefore be "balanced".

The second set of features defined in [24] measure characteristics of the degrees of the endpoint vertices u and v of the given edge (u,v). There are five of these features, quantifying the positive and negative out-degrees of u, the positive and negative in-degrees of v, and the total number of neighbors u and v have in common (interpreted in an undirected sense). Combining these five measures with the sixteen triad-related features results in a feature vector $x\in\Re^{21}$ for each edge of interest (see [24] for a more thorough discussion of these features and the motivation for selecting them). The feature vector x associated with an edge (u,v) will form the basis for predicting the sign of that edge.

We wish to learn a vector $c\in\Re^{21}$ such that the classifier orient = sign($c^T x$) accurately estimates the sign of the edge whose features are encoded in vector x. Vector c is learned, in part, from labeled examples of positive and negative edges. Additionally, the proposed learning algorithm leverages the insights of SBT. A simple way to incorporate SBT is to assemble sets $V^+$ and $V^-$ of positive and negative features, that is, sets of features which according to SBT ought to be associated with positive and negative edges, respectively. The triads to which (u,v) belongs in which the other two edges are positive are predicted by SBT to "contribute" to (u,v) being positive; thus the four features corresponding to triads with two positive labeled edges are candidates for membership in $V^+$ (there are four such features because $G_s$ is directed). Analogously, SBT posits that the eight features indexing triads in which exactly one of the two edges that neighbor (u,v) is positive are candidates for membership in $V^-$. (Note that the remaining four triad features index triads in which both of the edges neighboring (u,v) are negative, and as there is less empirical support for SBT in this case [24] these features are not assigned to either $V^+$ or $V^-$.)

We now derive an ML algorithm for edge-sign prediction which is capable of leveraging SBT in its learning process. The development begins by modeling the problem data as a bipartite graph $G_b$ of edge-sign instances and features (see Figure 1). If there are n edges and 21 features, it can be seen that the adjacency matrix A for graph $G_b$ is given by

$$A = \begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix} \qquad (1)$$

where matrix $X\in\Re^{n\times 21}$ is constructed by stacking the n feature vectors $x_i$ as rows, and each '0' is a matrix of zeros.

Assume the initial problem data consists of a set of n edges, of which $n_l \le n$ are labeled, and a set of labeled features $V_l = V^+\cup V^-$, and suppose this label information is encoded as vectors $d\in\Re^{nl}$ and $w\in\Re^{|Vl|}$, respectively. Let $d_{est}\in\Re^n$ be the vector of estimated signs for the edges in the dataset, and define the "augmented" classifier $c_{aug} = [d_{est}^T \quad c^T]^T \in \Re^{n+21}$ that estimates the polarity of both edges and features. Note that the quantity $c_{aug}$ is introduced for notational convenience and is not directly employed for classification. More specifically, in the proposed methodology we learn $c_{aug}$, and therefore c, by solving an optimization problem involving the labeled and unlabeled training data, and then use c to estimate the sign of any new edge of interest with the simple classifier orient=sign($c^Tx$). Assume for ease of notation that the instances and features are indexed so the first $n_l$ elements of $d_{est}$ and $|V_l|$ elements of c correspond to labeled data.

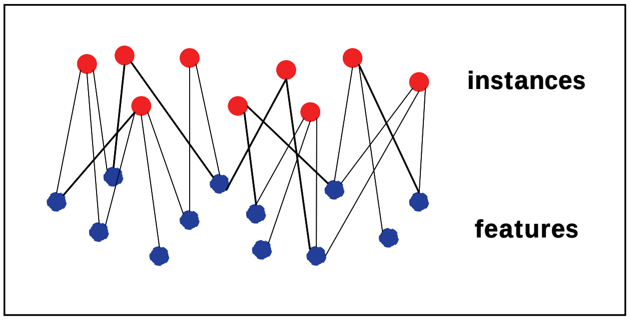


**Figure 1.** Cartoon of bipartite graph data model $G_b$, in which edge-instances (red vertices) are connected to the features (blue vertices) they contain, and link weights (black edges) reflect the magnitudes taken by the features in the associated instances.

We wish to learn an augmented classifier $c_{aug}$ with the following three properties: 1.) if an edge is labeled, then the corresponding entry of $d_{est}$ should be close to this ±1 label; 2.) if a feature is in the set $V_l = V^+ \cup V^-$, then the corresponding entry of c should be close to this ±1 polarity; and 3.) if there is an edge $X_{ij}$ of $G_b$ that connects an edge x and a feature f and $X_{ij}$ possesses significant weight, then the estimated polarities of x and f should be similar. These objectives are encoded in the following optimization problem:

$$\min_{c_{aug}} \; c_{aug}^T L c_{aug} + \beta_1 \sum_{i=1}^{n_l} (d_{est,i} - d_i)^2 + \beta_2 \sum_{i=1}^{|V_l|} (c_i - w_i)^2 \qquad (2)$$

where L = D − A is the graph Laplacian matrix for $G_b$, with D the diagonal degree matrix for A (i.e., $D_{ii} = \Sigma_j A_{ij}$), and $\beta_1$, $\beta_2$ are nonnegative constants. Minimizing (2) enforces the three properties we seek for $c_{aug}$, with the second and third terms penalizing "errors" in the first two properties. To see that the first term enforces the third property, observe that this expression is a sum of components of the form $X_{ij}(d_{est,i} - c_j)^2$. The constants $\beta_1$, $\beta_2$ are used to balance the relative importance of the three properties. Note that in situations where the set of available labeled instances is very limited, classifier performance often can be improved by replacing L in (2) with the normalized Laplacian $L_n$=$D^{-1/2}LD^{-1/2}$, or with a power of this matrix $L_n^k$ (for k a positive integer); this modification serves to "smooth" the polarity estimates assigned to the vertices of $G_b$.

The $c_{aug}$ that minimizes objective function (2) can be obtained by solving the following set of linear equations:

$$\begin{bmatrix} L_{11} + \beta_1 I_{nl} & L_{12} & L_{13} & L_{14} \\ L_{21} & L_{22} & L_{23} & L_{24} \\ L_{31} & L_{32} & L_{33} + \beta_2 I_{|V_l|} & L_{34} \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix} c_{aug} = \begin{bmatrix} \beta_1 d \\ 0 \\ \beta_2 w \\ 0 \end{bmatrix} \qquad (3)$$

where the $L_{ij}$ are matrix blocks of L of appropriate dimension.

We summarize this discussion by sketching an algorithm for learning the proposed edge-sign prediction (ESP) classifier:

**Algorithm ESP**

1. Construct the set of equations (3).
2. Solve equations (3) for $c_{aug} = [\, d_{est}^T \quad c^T \,]^T$ (for instance using the Conjugate Gradient method).
3. Estimate the sign of any new edge x of interest as: orient = sign($c^Tx$).

The utility of Algorithm ESP is now examined through a case study involving edge-sign estimation for two social networks extracted from the Wikipedia online encyclopedia.

### C. *Wikipedia Case Study*

This case study examines the performance of Algorithm ESP for the problem of estimating the signs of the edges in two social networks extracted from Wikipedia (WP), a collectively-authored online encyclopedia with an active user community. We consider the following WP social networks: 1.) the graph of 103,747 edges corresponding to votes cast by WP users in elections for promoting individuals to the role of 'admin' [24], and 2.) the graph of 740,397 edges characterizing editor interactions in WP [28]. In each network, the majority of the edges (≈80%) are positive. Thus we follow [24] and create balanced datasets consisting of 20K positive and 20K negative edges for the "voting" network [24], and 50K positive and 50K negative edges for the "interaction" network [28].

This study compares the edge-sign prediction accuracy of Algorithm ESP with that of the impressive gold-standard logistic regression classifier given in [24]. The gold-standard algorithm is applied exactly as described in [24]. Algorithm ESP is implemented with parameter values $\beta_1 = 0.1$ and $\beta_2 = 0.5$, and with vector w constructed using the four "positive triad" features $V^+$ and eight "negative triad" features $V^-$ defined above. As a focus of the investigation is evaluating the extent to which good prediction performance can be achieved even when only a limited number of labeled edges are available for training, we examine training sets which incorporate a range of numbers of labeled edges: $n_l$ = 0, 10, 20, 50, 100, 200.

Sample results from this study are depicted in Figures 2 and 3. Each data point in the plots represents the average of ten trials. In each trial, the edges are randomly split into equal-size training and testing sets, and a randomly selected subset of the training edges of size $n_l$ is "labeled" (i.e., the labels for these edges are made available to the learning algorithms). It can be seen that Algorithm ESP outperforms the gold-standard method on both datasets, and that the improved accuracy obtained with the proposed "SBT-informed" algorithm is particularly significantly when the number of labeled training instances is small. An interesting open question is the extent to which this ability to provide good performance with limited labeled data implies a similar robustness to erroneously labeled data. The accuracy of the proposed algorithm does not depend sensitively on parameters $\beta_1$, $\beta_2$, so that the method is convenient to apply.

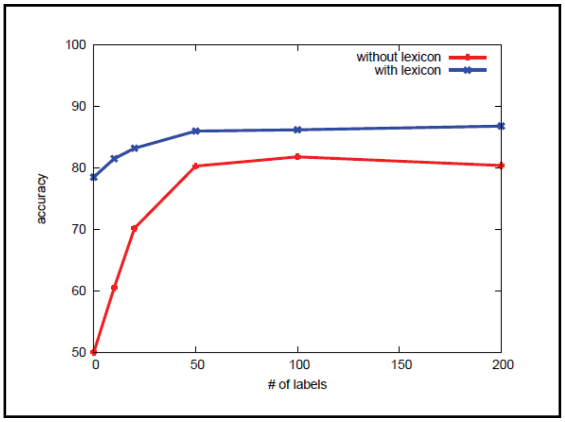


**Figure 2.** Results for WP "voting network" case study. The plot shows how edge-sign prediction accuracy (vertical axis) varies with the number of available labeled training instances (horizontal axis) for two classifiers: gold-standard (red) and Algorithm ESP (blue).

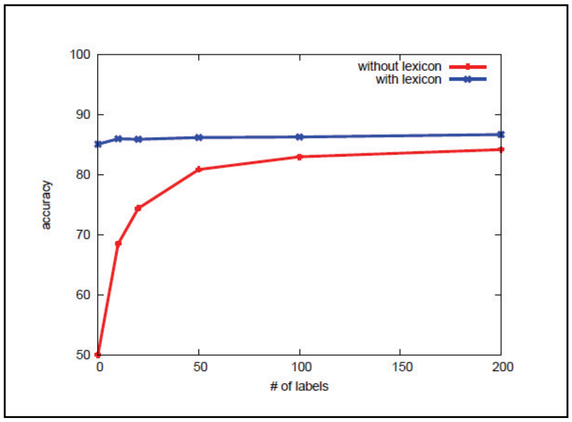


**Figure 3.** Results for WP "interaction network" case study. The plot shows how edge-sign prediction accuracy (vertical axis) varies with the number of available labeled training instances (horizontal axis) for two classifiers: gold-standard (red) and Algorithm ESP (blue).

### D. *Network Fracture Case Study*

Recently it has been proposed that structural balance theory can be used to predict the way a network of entities (e.g., individuals, countries) will split if subjected to stress [29], a capability of relevance in many security applications. Briefly, [29] models the polarity and intensity of relationships between the entities of interest as a completely connected network with weighted adjacency matrix $Z=Z^T\in\Re^{n\times n}$, where matrix element $z_{ij}$ represents the strength of the friendliness or unfriendliness between entities i and j. Note that this network model is somewhat more general than the one introduced above, in that each edge relating two individuals possesses both a sign and an intensity.

SBT is a "static" theory, positing what a stable configuration of edge-signs in a social network should look like. However, underlying the theory is a dynamical idea of how unbalanced network triads ought to resolve themselves to become balanced. A model which captures this underlying dynamics is given by the simple matrix differential equation [29]

$$dZ/dt = Z^2, \quad Z(0)=Z_0. \qquad (4)$$

To see the connection between these dynamics and SBT, observe that (4) specifies the following dynamics for entry $z_{ij}$:

$$dz_{ij}/dt = \Sigma_k \, z_{ik} \, z_{kj}.$$

Thus if triad {i,j,k} is such that $z_{ik}$ and $z_{kj}$ have the same sign, the participation of $z_{ij}$ in this triad will drive $z_{ij}$ in the positive direction, while if they have opposite signs then $z_{ij}$ will be driven in the negative direction. These dynamics therefore favor triads with an odd number of positive edge-signs, consistent with SBT [22].

The paper [29] proves that, for generic initial conditions $Z_0$, system (4) evolves to a balanced pattern of edge-signs in finite time; the balanced configuration is guaranteed to be composed of either all positive edges or two all-positive cliques connected entirely by negative edges. These configurations can be interpreted as predictions of the way a social network described by $Z_0$ will fracture if subjected to sufficient stress. More precisely, given a model $Z_0$ for a signed social network, model (4) can be used as the basis for the following two-step procedure for predicting the way the network will fracture: 1.) integrate (4) forward in time until it reaches singularity $Z_s$ (this singularity will be reached in finite time), and 2.) interpret $Z_s$ as defining a split of the network into two groups, where each group has all positive intra-group edges and the inter-group edges are all negative (and where one of the groups could be empty). See Figure 4 for an illustration of the dynamics of system (4).

Remarkably, [29] shows that predictions obtained in this manner are in excellent agreement with two real-world cases of group fracture for which there is empirical data: the division of countries into Allied and Axis powers in World War II [30], and the split of the well-studied Zachary Karate Club into two smaller clubs [31]. However, the analysis presented in [29] requires that matrix $Z_0$ be completely known, that is, that all of the "initial" relationships $z_{ij}(0)$ between entities be measurable. Such comprehensive data are not always available in practical applications.

We have found that the requirement that relationship matrix $Z_0$ be perfectly known can be relaxed through the use of Algorithm ESP. More specifically, given a subset of the relationship data, the remaining weighted edge-signs can be estimated using Algorithm ESP, and these estimates $\underline{Z}_0$ can be used in place of $Z_0$ when initializing (4). We have tested this procedure using the relationship network proposed in [30] for 17 key countries involved in World War II. This investigation demonstrates that accurate prediction of which countries would eventually join the Allied forces and which would become Axis members can be made with less than 15% of the edge-signs known in advance. For example, data for only the relationships maintained by Germany and the USSR is sufficient to enable correct prediction of the ultimate alignment of all countries except Portugal  (see Figure 4). Similar results hold for analysis of the split of the Zachary Karate Club [31].

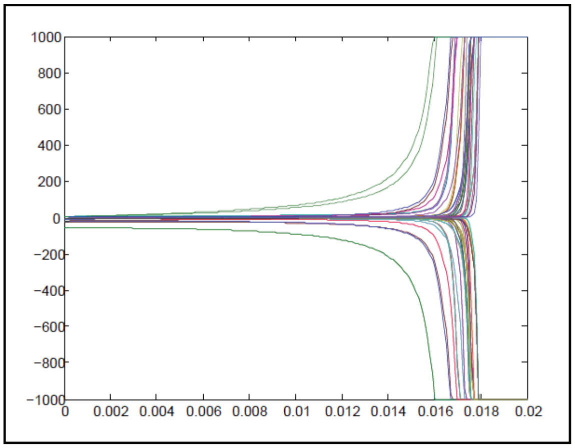


**Figure 4.** SBT dynamics. The evolution of model (4) initialized at the (scaled) "propensity" matrix given in [30] (horizontal axis is time and vertical axis is edge-weight).

## III. Summary

This two-part paper proposes that predictive analysis methods often can be improved by leveraging sociological models, and explores this possibility by considering two challenging prediction tasks: 1.) inferring signs (friendly or antagonistic) of ties in social networks, and 2.) predicting whether an emerging social diffusion event will propagate widely or quickly dissipate. In this first part of the paper, we derive a novel ML algorithm for edge-sign prediction which leverages structural balance theory [22-24]. The proposed algorithm outperforms a gold-standard method in empirical tests with large-scale online social networks, and the inferred edge-signs are shown to be useful when predicting the way adversarial networks are likely to fracture under stress.

Part Two of the paper examines the problem of forecasting the ultimate reach of "complex contagion" events [25,26], and develops a new "sociology-aware" ML algorithm for predicting which complex contagion events will ultimately propagate widely and which will quickly dissipate. Taken together, these results suggest that incorporating simple models from sociology can substantially improve the performance of prediction methods, particularly in applications in which there is limited data available for training and implementing the algorithms.

## Acknowledgements

This work was supported by the U.S. Department of Defense, The Boeing Company, and the Laboratory Directed Research and Development Program at Sandia National Laboratories. We thank Curtis Johnson and Travis Bauer of Sandia and Anne Kao of Boeing for numerous helpful discussions on aspects of this research.

## References

[1] Colbaugh, R. and K. Glass, "Predictive analysis for social processes I: Multi-scale hybrid system modeling, and II: Predictability and warning analysis", *Proc. 2009 IEEE Multi-Conference on Systems and Control*, Saint Petersburg, Russia, July 2009.

[2] Choi, H. and H. Varian, "Predicting the present with Google Trends", SSRN Preprint, April 2009.

[3] Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Second Edition, Springer, New York, 2009.

[4] Colbaugh, R., K. Glass, and P. Ormerod, "Predictability of 'unpredictable' cultural markets", *105th Annual Meeting of the American Sociological Association*, Atlanta, GA, August 2010.

[5] Asur, S. and B. Huberman, "Predicting the future with social media", *Proc. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Toronto, Ontario, Canada, September 2010.

[6] Goel, S., J. Hofman, S. Lahaie, D. Pennock, and D. Watts, "Predicting consumer behavior with Web search", *Proc. National Academy of Sciences USA*, Vol. 107, pp. 17486-17490, 2010.

[7] Lerman, K. and T. Hogg, "Using stochastic models to describe and predict social dynamics of Web users", arXiv preprint, October 2010.

[8] Bollen, J., H. Mao, and X. Zeng, "Twitter mood predicts the stock market", *J. Computational Science*, Vol. 2, pp. 1-8, 2011.

[9] Colbaugh, R. and K. Glass, "Detecting emerging topics and trends via predictive analysis of 'meme' dynamics", *Proc. 2011 AAAI Spring Symposium Series*, Palo Alto, CA, March 2011.

[10] Lui, C., P. Metaxas, and E. Mustafaraj, "On the predictability of the U.S. elections through search volume activity", *Proc. IADIS e-Society Conference*, Avila, Spain, March 2011.

[11] Amodea, G., R. Blanco, and U. Brefeld, "Hybrid models for future event prediction", *Proc. CIKM '11*, Glasgow, Scotland, October 2011.

[12] Colbaugh, R. and K. Glass, "Early warning analysis for social diffusion events", *Security Informatics*, accepted for publication.

[13] Clauset, A., C. Moore, and M. Newman, "Hierarchical structure and the prediction of missing links in networks", *Nature*, Vol. 453, pp. 98-101, 2008.

[14] Pang, B. and L. Lee, "Opinion mining and sentiment analysis", *Foundations and Trends in Information Retrieval*, Vol. 2 , pp. 1-135, 2008.

[15] Abbasi, A., H. Chen, and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums", *ACM Transactions on Information Systems*, Vol. 26, pp. 1-34, 2008.

[16] Lampos, V., T. De Bie, and N. Cristianini, "Flu detector – Tracking epidemics on Twitter, *ECML PKDD 2010*, Springer LNAI 6323, 2010.

[17] Christakis, N. and J. Fowler, "Social network sensors for early detection of contagious outbreaks", *PLoS ONE*, Vol. 5, e12948, 2010.

[18] Ayers, J., K. Ribisi, and J. Brownstein, "Tracking the rise in popularity of electronic nicotine delivery systems using search query surveillance", *American J. Preventative Medicine*, Vol. 41, pp. 1-6, 2011.

[19] Glass, K. and R. Colbaugh, “Estimating the sentiment of social media content for security informatics applications”, *Security Informatics*, Vol. 1, No. 3, pp. 1-16, 2012.

[20] Colbaugh, R. and K. Glass, “Proactive defense for evolving cyber threats”, *Proc. 2011 IEEE International Conference on Intelligence and Security Informatics*, Beijing, China, July 2011.

[21] Colbaugh, R., “Arctic ice, George Clooney, lipstick on a pig, and insomniac fruit flies: Combining kd and m&s for predictive analysis”, *Proc. ACM KDD ’11*, San Diego, CA, August 2011.

[22] Heider, F., “Attitude and cognitive organization”, *J. Psychology*, Vol. 21, pp. 107-112, 1946.

[23] Cartwright, D. and F. Harary, “Structural balance: A generalization of Heider’s theory”, *Psychological Review*, Vol. 63, pp. 277-293, 1956.

[24] Leskovec, J., D. Huttenlocher, and J. Kleinberg, “Predicting positive and negative links in online social networks”, *Proc WWW 2010*, Raleigh, NC, April 2010.

[25] Colbaugh, R. and K. Glass, “Leveraging sociological models for prediction II: Early warning for complex contagions”, *Proc. 2012 IEEE International Conference on Intelligence and Security Informatics*, Washington, DC USA, June 2012.

[26] Centola, D., “The spread of behavior in an online social network experiment”, *Science*, Vol. 329, pp. 1194-1197, 2010.

[27] Leskovec, J., L. Backstrom, and J. Kleinberg, “Meme-tracking and the dynamics of the news cycle”, *Proc. ACM KDD ‘09*, Paris, France, June 2009.

[28] Maniu, S., B. Cautis, and T. Abdessalem, “Building a signed network from interactions in Wikipedia”, *Proc. DBsocial ’11*, Athens, Greece, June 2011.

[29] Marvel, S., J. Kleinberg, R. Kleinberg, and S. Strogatz, “Continuous-time model of structural balance”, *Proc. National Academy of Sciences USA*, Vol. 108, pp. 1771-1776, 2011.

[30] Axelrod, R. and D. Bennett, “Landscape theory of aggregation”, *British J. Political Science*, Vol. 23, pp. 211-233, 1993.

[31] Zachary, W., “Information flow model for conflict and fission”, *J. Anthropological Research*, Vol. 33, pp. 452-473, 1977.

# Leveraging Sociological Models for Prediction II: Early Warning for Complex Contagions

Richard Colbaugh
Sandia National Laboratories
Albuquerque, NM USA
colbaugh@comcast.net

Kristin Glass
New Mexico Institute of Mining and Technology
Socorro, NM USA
kglass@icasa.nmt.edu

***Abstract*—There is considerable interest in developing techniques for predicting human behavior, and a promising approach to this problem is to collect phenomenon-relevant empirical data and then apply machine learning methods to these data to form predictions. This two-part paper shows that the performance of such learning algorithms often can be improved substantially by leveraging sociological models in their development and implementation. In this paper, the second of the two parts, we demonstrate that a sociologically-grounded learning algorithm outperforms a gold-standard method for the task of predicting whether nascent social diffusion events will "go viral". Significantly, the proposed algorithm performs well even when there is only limited time series data available for analysis.**

***Keywords*—**predictive analysis, sociological models, social networks, machine learning.

## I. INTRODUCTION

There is great interest in developing techniques for accurately predicting human behavior. For example, forecasting the eventual outcomes of social processes is a central concern in domains ranging from popular culture to public policy to national security [1]. The task of inferring the existence and nature of activities which are presently underway but not directly observable, sometimes referred to as "predicting the present" [2], is also of crucial importance in many applications. A promising approach to obtaining such predictions is to collect empirical data which appropriately characterize the phenomenon of interest and then to analyze these data using machine learning (ML) methods [3]. Existing ML techniques, although useful, face at least two key challenges: 1.) the prediction accuracy obtainable even with state-of-the-art algorithms is sometimes insufficient for the task at hand, such as when the predictions are to be used to inform high-consequence decisions, and 2.) applying ML methods typically requires that significant quantities of data be collected and "labeled" for use in algorithm training.

This two-part paper proposes that the challenges of predicting human behavior using ML often can be overcome by leveraging sociological models in the development and implementation of the learning algorithms. Part One of the paper considers the problem of predicting the "signs" of relationships in social networks, where positive and negative edges reflect friendly and antagonistic social ties, respectively, and derives a novel ML algorithm for edge-sign prediction that is based in part on structural balance theory [4]. In the present paper, the second of the two parts, we examine the problem of forecasting the ultimate reach of "complex contagion" events [5,6]. Predictability assessment of complex contagion dynamics indicates that the metrics which should be predictive of the contagion's reach are fairly subtle measures of the network dynamics associated with early diffusion activity. These results are used to derive an ML algorithm for predicting which complex contagions will ultimately "go viral" and which won't, and it is demonstrated that the algorithm outperforms gold-standard methods in an empirical investigation of online meme propagation [7]. Significantly, the new algorithm performs well even when only limited time series data are available for analysis, permitting reliable predictions early in the contagion lifecycle. We also show that the proposed algorithm enables effective early warning analysis for an important class of cyber threats.

## II. EARLY WARNING FOR COMPLEX CONTAGIONS

### A. Problem Formulation

There is significant interest in developing predictive capabilities for social diffusion processes, for instance to permit early identification of emerging contentious situations or accurate forecasting of the eventual reach of potentially "viral" behaviors. This section considers the following early warning problem: we suppose some sort of triggering event has taken place and wish to determine, as early as possible, whether this event will ultimately generate a large, self-sustaining reaction, involving the propagation of behavioral changes through a substantial portion of a population, or will instead quickly dissipate. Of particular interest is propagation of behaviors that are costly or controversial, or about which there is uncertainty, as these activities often have large security-relevant impacts [4].

Recent research has shown that such behaviors may spread as *complex contagions*, requiring social affirmation or reinforcement from multiple sources in order to propagate [5,6]. Because the diffusion dynamics for complex contagions are different than those of "simple" contagions like disease epidemics, it is natural to suspect that developing effective early warning algorithms for complex contagions may require careful consideration of these more complex dynamics. In this section we explore this possibility by deriving an early warning method for complex contagions which explicitly leverages a mathematical model for these diffusion events. We adopt the contagion model proposed in [6], implemented on a class of social networks which possess realistic topologies, and analyze this model to identify features of the contagion that are likely to be predictive of diffusion reach. These features are then used as the basis for an ML algorithm which distinguishes complex contagions that will propagate widely from those which will quickly dissipate.

### *B. Predictability Assessment*

Here we briefly describe the results of applying the predictability assessment procedure presented in [1] to the task of identifying measurables that should be predictive of complex contagion success. The discussion begins with short, intuitive reviews of our predictability assessment process and network diffusion modeling framework, and then summarizes the main results obtained via this theoretical analysis.

**Predictability.** The basic idea behind the proposed approach to predictability analysis is simple and natural: we assess predictability by answering questions about the reachability of diffusion events. To obtain a mathematical formulation of this strategy, the behavior about which predictions are to be made is used to define the system *state space subsets of interest* (SSI), while the particular set of candidate measurables under consideration allows identification of the *candidate starting set* (CSS), that is, the set of states and system parameter values which represent initializations that are consistent with, and equivalent under, the presumed observational capability. As a simple example, consider an online market with two products, A and B, and suppose the system state variables consist of the current market share for A, ms(A), and the rate of change of this market share, r(A) (ms(B) and r(B) are not independent state variables because ms(A) + ms(B) = 1 and r(A) + r(B) = 0); let the parameters be the advertising budgets for the products, b(A) and b(B). The producer of A might find it useful to define the SSI to reflect market share dominance by A, that is, the subset of the two-dimensional state space where ms(A) exceeds a specified threshold. If only market share and the advertising budgets can be measured then the CSS is the one-dimensional subset of state-parameter space consisting of the initial magnitudes for ms(A), b(A), and b(B), with r(A) unspecified.

Roughly speaking, the approach to predictability assessment proposed in [1] involves determining how probable it is to reach the SSI from a CSS and deciding if these reachability properties are compatible with the prediction goals. If a system's reachability characteristics are compatible with the prediction objectives the situation is deemed predictable, and otherwise it is unpredictable. This setup permits the identification of candidate predictive measurables: these are the measurable states and/or parameters which most strongly affect the predictability properties [1]. Continuing with the online market example, if trajectories with positive early market share rates r(A) are much more likely to yield market share dominance for A than are trajectories with negative early r(A), independent of the early values for ms(A), then the situation is unpredictable (because r(A) is not measured). Adding the capacity to measure r(A) would then increase system predictability, and depending upon the task requirements this new measurement ability could result in a predictable situation. A quantitative, mathematically-rigorous presentation of this predictability assessment framework can be found in [1].

**Model.** In complex contagion events, the probability of adopting a controversial or unproven behavior or idea increases with the number of other adopting *individuals*, and not merely the number of exposures to the contagion (so that multiple interactions with the same adopting individual do not increase the likelihood of adoption, as it does in simple contagions) [5,6]. Recently the authors of [6] proposed an empirically-grounded model for complex contagions in which individuals interact via a social network of arbitrary topology, and the probability that individual A adopts a given activity or idea is a function of the number of A's adopting neighbors; the functional form of this adoption "influence curve" is obtained empirically (see [6] for a detailed description of the model).

The dynamics of contagion may depend upon the topological structure of the underlying social network. This dependence suggests that, in order to identify the features of complex contagions which have predictive power, it is necessary to assess predictability using social network models with realistic topologies. Therefore in this study we implement the complex contagion model [6] with social networks that possess four topological properties which are ubiquitous in the real-world [1]: right-skewed degree distribution, transitivity, community structure, and core-periphery structure.

It is shown in [1] that *stochastic hybrid dynamical systems* (S-HDS) provide a useful mathematical formalism with which to represent social contagions on realistic networks (see Figure 1). An S-HDS is a feedback interconnection of a discrete-state stochastic process, such as a Markov chain, with a family of continuous-state stochastic dynamical systems [1]. Combining discrete and continuous dynamics within a unified, computationally tractable framework offers an expressive, scalable modeling environment that is amenable to formal mathematical analysis. In particular, S-HDS models can be used to efficiently represent and analyze social contagion on large-scale networks with the four topological properties listed above [1].

As an intuitive illustration of the way S-HDS enable effective, tractable representation of complex contagion phenomena, consider the task of modeling contagion on a network possessing community structure. As shown in Figure 1, the contagion proceeds in two ways: 1.) *intra-community diffusion*, involving frequent interactions between individuals within the same community and the resulting gradual change in the concentrations of adopting (red) individuals, and 2.) *inter-community diffusion*, in which the "infection" jumps from one community to another, for instance because an adopting individual encounters a new community. S-HDS models offer a natural framework for representing these dynamics, with the S-HDS continuous system modeling the intra-community dynamics (e.g., via stochastic differential equations), the discrete system capturing inter-community dynamics (e.g., using a Markov chain), and the interplay between these dynamics being encoded in the S-HDS feedback structure (e.g., the transition probabilities of the discrete system Markov chain may depend upon the state of the continuous system) [1].

**Results.** We applied the predictability assessment methodology summarized above to a "realistic network" version of the complex contagion model given in [6] (i.e., the model obtained by implementing the dynamics specified in [6] on a class of networks possessing the four topological properties summarized above). The main finding of this study is that the predictability of the reach of complex contagions depends crucially upon the social network's community and core-periphery structures. These findings are now summarized more quantitatively.

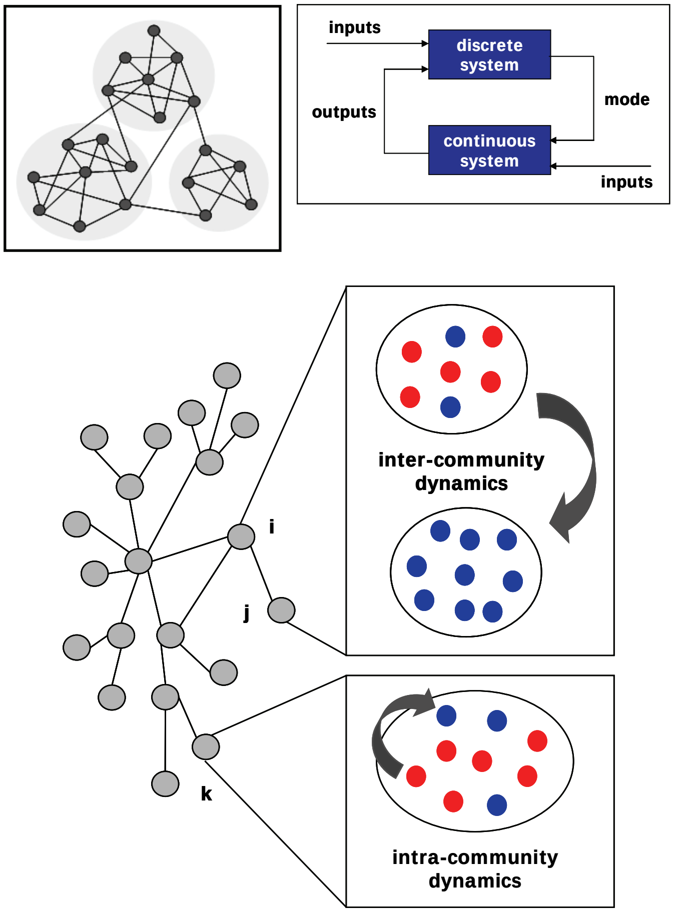


**Figure 1.** Modeling complex contagions on networks with community structure via S-HDS. The cartoon at top left depicts a network with three communities. The cartoon at bottom illustrates contagion *within* a community k and *between* communities i and j. The schematic at top right shows the basic S-HDS feedback structure.

We adopt a modularity-based definition for network community structure [8], whereby a good partitioning of a network's vertices into communities is one for which the number of edges between putative communities is smaller than would be expected in a random partitioning. To be concrete, a modularity-based partitioning of a network into two communities maximizes the modularity $Q = s^T B s / 4m$, where m is the total number of edges in the network, the partition is specified with the elements of vector s by setting $s_i = 1$ if vertex i belongs to community 1 and $s_i = -1$ if it belongs to community 2, and the matrix B has elements $B_{ij} = A_{ij} - k_i k_j / 2m$, with $A_{ij}$ and $k_i$ denoting the network adjacency matrix and degree of vertex i, respectively. Partitions of the network into more than two communities can be constructed recursively [8]. This definition enables the specification of the first candidate predictive feature nominated by our predictability assessment: early dispersion of a complex contagion process across network communities should be a reliable predictor that the ultimate reach of the contagion will be significant.

We characterize network core-periphery structure in terms of the k-shell decomposition [9]. To partition a network into its k-shells, one first removes all vertices with degree one, repeating this step if necessary until all remaining vertices have degree two or higher; the removed vertices constitute the 1-shell. Continuing in the same way, all vertices with degree two (or less) are recursively removed, creating the 2-shell. This process is repeated until all vertices have been assigned to a k-shell, and the shell with the highest index, the $k_{max}$-shell, is deemed to be the core of the network. This definition permits us to state the second candidate predictive feature nominated via theoretical predictability assessment: early contagion activity within the network $k_{max}$-shell should be a reliable predictor that the reach of the diffusion will be significant.

### C. *Prediction Algorithm*

Consider the problem of predicting, very early in the lifecycle of a complex contagion event, whether or not the contagion will propagate widely. We adopt an ML approach to this early warning task: given a triggering incident, one or more information sources which reflect the reaction to this trigger by a population of interest, and a specification for what constitutes an "alarming" reaction, the goal is to learn a classifier that accurately predicts, as early as possible, whether or not reaction to the event will eventually become alarming. The ML classifier used in this investigation is the Avatar ensembles of decision trees (A-EDT) algorithm [10]; qualitatively similar results were obtained in tests with other, less sophisticated classifiers [3].

A key step in early warning analysis is determining which characteristics of the phenomenon of interest, if any, possess exploitable predictive power. Based on the results of the preceding predictability assessment study, we consider three general classes of features: 1.) *intrinsics-based features* – measures of the inherent properties and attributes of the "object" being diffused, 2.) *simple dynamics-based features* – metrics which capturing simple properties of the diffusion dynamics (e.g., the rate at which the diffusion is propagating), 3.) *network dynamics-based features* – measures that characterize the way the early diffusion is progressing relative to the network's community and core-periphery structures. Precise definitions for the features in these classes are, of course, application dependent.

The proposed approach to early warning analysis is to identify and collect features from these classes for the event of interest, input the feature values to the A-EDT classifier, and then run the classifier to generate a warning prediction (i.e., a forecast that the event is expected to become 'alarming' or remain 'not alarming'). The algorithm presented below specifies this procedure in general terms, and illustrative instantiations of the process are given in the case studies discussed in Sections IID and IIE. It is assumed that social media data form the primary source of information concerning events of interest [11]. However, the analysis is very similar when alternative sources of data are employed [1].

Consider the following early warning algorithm:

**Algorithm EW**

Given: a triggering incident, a definition for what constitutes an ‘alarming’ reaction, and a set of social media sites (e.g., blogs) B which are relevant to the early warning task.

Initialization: train the A-EDT classifier on a set of events that are qualitatively similar to the triggering event of interest and are labeled as ‘alarming’ or ‘not alarming’.

Procedure:

1. Assemble a lexicon of keywords L that pertain to the triggering event under study.
2. Conduct a sequence of Web crawls and construct a time series of blog graphs $G_B(t)$. For each time period t, label each blog in $G_B(t)$ as ‘active’ if it contains a post mentioning any of the keyword in L and ‘inactive’ otherwise.
3. Form the union $G_B = \cup_t G_B(t)$, partition $G_B$ into network communities and into k-shells, and map the partition element structure of $G_B$ back to each of the graphs $G_B(t)$.
4. For each graph $G_B(t)$, compute the values for all features (intrinsics, simple dynamics, and network dynamics).
5. Apply the A-EDT classifier to the time series of features, i.e., the features obtained for the sequence of blog graphs $\{G_B(t_0), \ldots, G_B(t_p)\}$, where $t_0$ and $t_p$ are the triggering event time and present time, respectively. Issue a warning alert if the classifier output is ‘alarming’.

We now offer a few remarks concerning Algorithm EW. The keywords in Step 1 can be identified with the help of subject matter experts and also through computational means (e.g., via meme analysis [1]). Step 2 is by now standard, and a variety of tools exist which can perform these tasks [11]. In Step 3, the blog network can be partitioned into communities and k-shells using modularity-based community extraction [8] and standard k-shell decomposition [9], respectively. The particular choices of metrics for the intrinsics, simple dynamics, and network dynamics features computed in Step 4 tend to be problem specific, and typical examples are given in the case studies below. Finally, in Step 5 the feature values obtained in Step 4 serve as inputs to the A-EDT classifier, and the output of the classifier is used to decide whether an alert should be issued.

### *D. Meme Case Study*

The goal of this case study is to apply Algorithm EW to the task of predicting whether or not a given *meme* (i.e., short textual phrase which propagates relatively unchanged online) will “go viral”. Although it may seem that meme diffusion is not sufficiently costly or controversial to qualify as a complex contagion, [6] shows that *political* memes appear to propagate in this way. Our main source of data on meme dynamics is the dataset archived at the site http://memetracker.org [12] by the authors of [7]. Briefly, the archive [12] contains time series data characterizing the online diffusion of ~70,000 memes during the period between 1 August and 31 December 2008. We are interested in using Algorithm EW to distinguish successful and unsuccessful political memes early in their lifecycle. More precisely, the prediction task is to classify memes into two groups – those which will ultimately be successful (acquire more than S posts) and those that will be unsuccessful (attract fewer than U posts) – very early in the meme lifecycle.

To support an empirical evaluation of the utility of Algorithm EW for this problem, we downloaded from [12] the time series data for slightly more than 70,000 memes. These data contain, for each meme M, a sequence of pairs $(t_1, URL_1)_M$, $(t_2, URL_2)_M$, …, $(t_T, URL_T)_M$, where $t_k$ is the time of appearance of the kth blog post or news article that contains at least one mention of meme M, $URL_k$ is the URL of the blog or news site on which that post/article was published, and T is the total number of posts that mention meme M. From this set of time series we randomly selected 100 “successful” political meme trajectories, defined as those corresponding to memes which attracted at least 1000 posts during their lifetimes, and 100 “unsuccessful” political meme trajectories, defined as those whose memes acquired no more than 100 total posts.

Two other forms of data were collected for this study: 1.) a large Web graph which includes websites (URLs) that appear in the meme time series, and 2.) samples of the text surrounding the memes in the posts which contain them. More specifically, we sampled the URLs appearing in the time series for our set of 200 successful and unsuccessful memes and performed a Web crawl that employed these URLs as “seeds”. This procedure generated a Web graph, denoted $G_B$, that consists of approximately 550,000 vertices (websites) and 1.4 million edges (hyperlinks), and includes essentially all of the websites which appear in the meme time series. To obtain samples of text surrounding memes in posts, we randomly selected ten posts for each meme and then extracted from each post the paragraph which contains the first mention of the meme.

Algorithm EW employs three types of features: intrinsics, simple dynamics-based, and network dynamics-based. We now describe the instantiation of each of these feature classes for the meme problem. Consider first the intrinsics features, which for the meme application become language-based measures. Each “document” of text surrounding a meme in its (sample) posts is represented by a simple “bag of words” feature vector $x \in \Re^{|V|}$, where the entries of x are the frequencies with which the words in the vocabulary V appear in the document. A language-based feature which might reasonably be expected to be predictive of meme propagation is the sentiment or emotion of documents containing the meme. A simple way to quantify a document’s sentiment/emotion is through the use of appropriate lexicons. Let $s \in \Re^{|V|}$ denote a lexicon vector, in which each entry of s is a numerical “score” quantifying the sentiment/emotion intensity of the corresponding word in vocabulary V. The aggregate sentiment/emotion score of document x can then be computed as $\text{score}(x) = s^T x / s^T 1$, where 1 is a vector of ones. Thus score(.) estimates document sentiment or emotion as a weighted average of the sentiment or emotion scores for the words comprising the document. (Note that if no sentiment or emotion information is available for a particular word in V then the corresponding entry of s is set to zero.)

To characterize the emotion content of a document we use the Affective Norms for English Words (ANEW) lexicon [13], while positive or negative sentiment is quantified via the “IBM lexicon” [14]. This approach generates four language features for each meme: the happiness, arousal, dominance, and positive/negative sentiment of the sample text surrounding that meme. As a preliminary test, we computed the mean emotion and sentiment of text surrounding the 100 successful and 100 unsuccessful memes in our dataset. On average the text surrounding successful memes is happier, more active, more dominant, and more positive than that surrounding unsuccessful memes ($p<0.0001$), so it is at least plausible that the language features may possess some predictive power.

Consider next two simple dynamics-based features, defined to capture basic characteristics of the early evolution of meme post volume: 1.) #posts($\tau$) – the cumulative number of posts mentioning the given meme by time $\tau$ (where $\tau$ is small relative to the typical meme lifespan), and 2.) post rate($\tau$) – a simple estimate of the rate of accumulation of these posts at time $\tau$. Recall that predictability assessment suggests that both early dispersion of contagion activity across network communities and early contagion activity within the network core ought to be predictive of meme success. These insights motivate the definition of two network dynamics-based features for meme prediction: 1.) community dispersion($\tau$) – the cumulative number of network communities in the blog graph $G_B$ that, by time $\tau$, contain at least one post which mentions the meme, and 2.) #k-core blogs($\tau$) – the cumulative number of blogs in the $k_{max}$-shell of blog graph $G_B$ that, by time $\tau$, contain at least one post which mentions the meme.

This case study compares the meme early warning accuracy of Algorithm EW, as applied to meme prediction, with that of two other prediction methods: a language-based (LB) strategy and a standard-dynamics (SD) scheme. The LB predictor uses the four language features noted above with the A-EDT classifier to try to distinguish successful and unsuccessful memes, and achieves a prediction accuracy of 66.5% (ten-fold cross-validation). Since simply guessing ‘successful’ for all memes gives an accuracy of 50%, it is seen that the language intrinsics, when used alone, possess relatively limited predictive power.

Next we compare the predictive performance of the SD classifier with that of Algorithm EW. The SD predictor combines the four language features with the two simple dynamics features, #posts($\tau$) and post rate($\tau$), within the A-EDT classifier. Because this is representative of state-of-the-art prediction schemes, this approach is referred to as the gold-standard algorithm. The application of Algorithm EW to meme prediction combines the language features with four dynamics measures: #posts($\tau$), post rate($\tau$), community dispersion($\tau$), and #k-core blogs($\tau$). Sample results from this empirical test are depicted in Figure 2. Each data point represents the average accuracy over ten trials (ten-fold cross-validation). It can be seen from Figure 2 that Algorithm EW outperforms the gold-standard method, especially in the important situation in which it is desired to form predictions soon after the meme is detected. Indeed, these results show that useful predictions can be obtained with Algorithm EW *within the first twelve hours* after a meme is detected (this corresponds to 0.5% of the average meme lifespan). Interestingly, analysis of feature predictive power [3] shows that the most predictive features are, in decreasing order, 1.) community dispersion, 2.) #k-core blogs, 3.) #posts, and 4.) post rate, which supports the conclusions of the complex contagion-based predictability assessment.

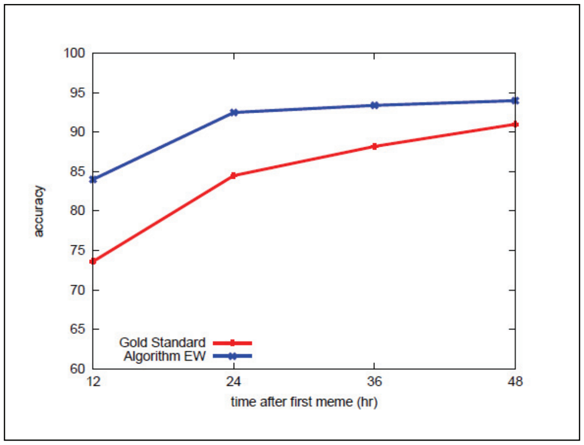


**Figure 2.** Results for meme early warning case study. The plot shows how prediction accuracy (vertical axis) varies with the length of time that has elapsed between meme detection and meme prediction (horizontal axis) for the two classifiers: gold-standard (red) and Algorithm EW (blue).

### *E. Cyber Early Warning Case Study*

This case study explores the ability of Algorithm EW to provide reliable early warning for politically-motivated distributed denial-of-service (DDoS) attacks, an important class of cyber threats. In particular, we are interested in exploring the utility of Algorithm EW when using social media as an information source. Toward this end, we first identified a set of Internet disruptions which included examples from three distinct classes of activity: 1.) successful DDoS attacks (the events for which we seek early warning; 2.) natural events which disrupt Internet service but for which it is known that no early warning signal exists in social media (e.g., earthquakes); 3.) quiet periods during which there is social media “chatter” concerning impending DDoS attacks but no successful attacks actually occurred. Including events selected from these three classes is intended to provide a fairly comprehensive test, as these classes correspond to 1.) the domain of interest, 2.) a set of disruptions which impact the Internet but have no social media warning signal, and 3.) a set of “non-events” which do not impact the Internet but do possess putative social media warning signals.

We selected twenty events from these three classes:

Politically-motivated DDoS attacks:

- Estonia event in April 2007;
- CNN/China incident in April 2008;
- Israel/Palestine conflict event in January 2009;
- DDoS associated with Iranian elections in June 2009;
- WikiLeaks event in November 2010;
- Anonymous v. PayPal, etc. attack in December 2010;
- Anonymous v. HBGary attack in February 2011.

Natural disturbances:

- European power outage in November 2006;

- Taiwan earthquake in December 2006;
- Hurricane Ike in September 2008;
- Mediterranean cable cut in January 2009;
- Taiwan earthquake in March 2010;
- Japan earthquake in March 2011.

Quiet periods:

Seven periods, from 2005 through 2011, during which there were discussions in social media of DDoS attacks on various U.S. government agencies but no successful attacks took place.

We collected two forms of data for each of these twenty events: *cyber data* and *social data*. The cyber data consist of time series of routing updates which were issued by Internet routers during a one month period surrounding each event. More precisely, these data are the Border Gateway Protocol (BGP) routing updates exchanged between gateway hosts in the Autonomous System network of the Internet. The data were downloaded from the publicly-accessible RIPE collection site [15] using the process described in [16]. The temporal evolution of the volume of BGP routing updates (e.g., withdrawal messages) gives a coarse-grained measure of the timing and magnitude of large Internet disruptions and thus offers a simple and objective way to characterize the impact of each of the events in our collection. The social data consist of time series of social media mentions of cyber attack-related keywords and Internet disruption-related keywords that were detected during a two month period surrounding each of the twenty events (in each case, event time was inferred from BGP data [16]). These data were gathered using the procedure specified in Algorithm EW.

We apply Algorithm EW to the task of distinguishing the seven DDoS attacks from the thirteen other events in the event set. For simplicity, in this case study we do not use any intrinsics-based features (e.g., language metrics) in the A-EDT classifier, and instead rely upon the four dynamics-based features defined in the meme study. We estimate the accuracy of Algorithm EW with two-fold cross-validation. In the case of DDoS events, the blog data made available to Algorithm EW is limited to posts made during the five week period which ended one week before the attack. For the six natural disturbances, the blog data includes all posts collected during the six week period immediately prior to the event, while in the case of the seven non-events, the blog data includes the posts gathered during a six week interval which spans discussions of DDoS attacks on U.S. government agencies.

In this evaluation, Algorithm EW achieves *perfect* accuracy, correctly identifying all ‘attack’ and ‘non-attack’ events. If the test is made more difficult, so that the blog data made available to Algorithm EW for attack events is limited to a four week period that ends two weeks before each attack, the proposed approach still achieves 95% accuracy, An examination of the predictive power of the four features used as inputs to the A-EDT classifier reveals that community dispersion is the most predictive measure.

## III. Summary

This two-part paper considers the challenging problem of predicting human behavior, and shows that incorporating simple models from sociology can substantially improve the performance of machine learning prediction methods, particularly in applications for which there is limited data available for training and implementing the algorithms. Future work will include investigating the predictability of the actions of opponents in adversarial settings through a combination of ML and sociologically-grounded game-theoretic models.

## Acknowledgements

This work was supported by the U.S. Department of Defense, The Boeing Company, and the Laboratory Directed Research and Development Program at Sandia National Laboratories. We thank Curtis Johnson of Sandia and Anne Kao of Boeing for numerous helpful discussions on aspects of this research.

## References

[1] Colbaugh, R. and K. Glass, “Early warning analysis for social diffusion events”, *Security Informatics*, accepted for publication.

[2] Choi, H. and H. Varian, “Predicting the present with Google Trends”, SSRN Preprint, April 2009.

[3] Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Second Edition, Springer, New York, 2009.

[4] Colbaugh, R. and K. Glass, “Leveraging sociological models for prediction I: Inferring adversarial relationships”, *Proc. 2012 IEEE International Conference on Intelligence and Security Informatics*, Washington, DC USA, June 2012.

[5] Centola, D., “The spread of behavior in an online social network experiment”, *Science*, Vol. 329, pp. 1194-1197, 2010.

[6] Romero, D., B. Meeder, and J. Kleinberg, “Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on Twitter”, *Proc WWW 2011*, Hyderabad, India, March 2011.

[7] Leskovec, J., L. Backstrom, and J. Kleinberg, “Meme-tracking and the dynamics of the news cycle”, *Proc. ACM KDD ‘09*, Paris, France, June 2009.

[8] Newman, M., “Modularity and community structure in networks”, *Proc. National Academy of Sciences USA*, Vol. 103, pp. 8577-8582, 2006.

[9] Carmi, S., S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir, “A model of Internet topology using the k-shell decomposition”, *Proc. National Academy of Sciences USA*, Vol. 104, pp. 11150-11154, 2007.

[10] http://www.sandia.gov/avatar/, accessed July 2010.

[11] Glass, K. and R. Colbaugh, “Web analytics for security informatics”, *Proc. 2011 European Intelligence and Security Informatics Conference*, Athens, Greece, September 2011.

[12] http://memetracker.org, accessed January 2010.

[13] Bradley, M. and P. Lang, “Affective norms for English words (ANEW): Stimuli, instruction manual, and affective ratings”, Technical Report C1, University of Florida, 1999.

[14] Ramakrishnan, G., A. Jadhav, A. Joshi, S. Chakrabarti, and P. Bhattacharyya, “Question answering via Bayesian inference on lexical relations”, *Proc. Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, July 2003.

[15] http://data.ris.ripe.net/, last accessed July 2011.

[16] Glass, K., R. Colbaugh, and M. Planck, “Automatically identifying the sources of large Internet events”, *Proc. IEEE International Conference on Intelligence and Security Informatics*, Vancouver, BC Canada, May 2010.

# Predictive Defense Against Evolving Adversaries

Richard Colbaugh
Sandia National Laboratories
Albuquerque, NM USA
colbaugh@comcast.net

Kristin Glass
New Mexico Institute of Mining and Technology
Socorro, NM USA
kglass@icasa.nmt.edu

***Abstract*—Adaptive adversaries are a primary concern in several domains, including cyber defense, border security, counterterrorism, and fraud prevention, and consequently there is great interest in developing defenses that maintain their effectiveness in the presence of evolving adversary strategies and tactics. This paper leverages the coevolutionary relationship between attackers and defenders to derive two new approaches to *predictive* defense, in which future attack techniques are anticipated and these insights are incorporated into defense designs. The first method combines game theory with machine learning to model and predict future adversary actions in the learner's "feature space"; these predictions form the basis for synthesizing robust defenses. The second approach to predictive defense involves extrapolating the evolution of defense configurations forward in time, in the space of defense parameterizations, as a way of generating defenses which work well against evolving threats. Case studies with a large cyber security dataset assembled for this investigation demonstrate that each method provides effective, scalable defense against current and future attacks, outperforming gold-standard techniques. Additionally, preliminary tests indicate that a simple variant of the proposed design methodology yields defenses which are difficult for adversaries to reverse-engineer.**

***Keywords*—**-predictive analytics, adversarial coevolution, machine learning, game theory, cyber security, security informatics.

## I. Introduction

Adaptive adversaries are a primary concern in many domains, including cyber defense, border security, counterterrorism, and crime prevention [e.g. 1-3]. For instance, emerging technologies and operational practices in these domains are increasingly moving toward highly interconnected architectures with small numbers of widely-shared protocols, thereby dramatically increasing the potential impact of even a single unanticipated attack. It is therefore essential that security professionals develop defenses which are able to respond rapidly to, or even foresee, evolving attack strategies and tactics.

Recognizing these trends and challenges, several researchers have recently proposed defenses which incorporate models of adversary behavior in order to increase defense system reliability and responsiveness against adaptive opponents; applications receiving attention include cyber defense [e.g. 4-7], border and transportation security [e.g. 8-10], and improvised explosive device defense [11,12]. However, while these model-informed methods represent an important advance over standard techniques, they continue to produce reactive defense designs and thus are limited in their ability to defend against new attacks.

Very recently, security researchers have begun working to develop *predictive* defenses, in which future attack strategies are explicitly anticipated and preemptively countered [13-16]. Despite this attention, much remains to be done to place the objective of predictive defense on a scientifically-grounded and practically-implementable foundation. Fundamental issues associated with the dynamics and predictability of coevolutionary "arms races" between attackers and defenders have yet to be resolved. For instance, although the work [13-15] has demonstrated that previous attacker actions and defender responses provide predictive information about future attacker behavior, little is known about which system characteristics have predictive power or how to employ these features to form useful predictions. Moreover, even in settings where these predictability and prediction issues have been resolved, it often remains an open question how to incorporate such predictive analytics into the design of practical real-world defense systems.

This paper leverages the coevolutionary relationship between attackers and defenders to derive two predictive defense algorithms which are effective against both current and future attacks strategies. We formulate the defense task as one of behavior classification, in which innocent and malicious activities are to be distinguished, and assume only limited historical information is available regarding prior attacker behavior or attack attributes. The first method combines game theory [17] with machine learning (ML) [18] to model and predict adversary actions in "feature space", that is, in the space of observable variables that the ML algorithm uses for learning; these predictions form the basis for synthesizing robust defenses. The second predictive defense strategy involves extrapolating the evolution of defense system configurations forward in time, in the space of defense parameterizations, as a way of generating defenses which work well against evolving threats. Interestingly, formulating the attack prediction/defense synthesis problem in an abstract space (of ML features or defense parameters) enables the development of algorithms that are scalable to applications of real-world size and complexity.

To permit the performance of these methods to be evaluated, we have assembled a large collection of non-Spam and Spam emails reflecting the evolution of Spammer tactics over an eight year period. Case studies with this dataset demonstrate that each of the proposed predictive methods provides robust, scalable defense, outperforming gold-standard Spam filters. Additionally, preliminary tests suggest that a simple "randomized feature" variant of the proposed design methodology generates defenses which are difficult for adversaries to reverse-engineer.

## II. Predictive Defense Via Game-Based Learning

### A. Problem Formulation

As indicated in the Introduction, there is significant interest in developing *predictive* approaches to defending against adaptive adversaries, in which opponents' evolving strategies are anticipated and these insights are employed to counter new attacks. This section considers the following concrete instantiation of the predictive defense problem: given some history of attacker actions, design a defense system which performs well against both current and future attacks.

It is reasonable to expect that concepts and techniques from game theory might be helpful in understanding adversary coevolution, and indeed such approaches have been explored in a variety of domains [5,10,19]. These investigations have revealed several challenges to successfully using game-theoretic methods for predictive defense, and we mention two that have been particularly daunting. First, the space of possible attacker actions is typically very large in realistic environments, and because the complexity of most game models increases exponentially with the number of actions available to players, this has often made these models intractable in practice [19]. Second, it has proved difficult to derive models that capture evolving attacker behavior in any but the most idealized situations.

We overcome these two challenges by developing a game-based model for adversary adaptation within an ML framework, enabling effective defense in realistic settings. Crucially, the proposed approach seeks to derive the optimal defense for new attacks, rather than to predict these attacks perfectly, and therefore enjoys robust performance in the presence of (inevitable) prediction errors. We approach the task of countering adversarial behavior as an ML classification problem, in which the objective is to distinguish innocent and malicious activity. Each instance of activity is represented as a feature vector $x \in \Re^{|F|}$, where entry $x_i$ of x is the value of feature i for this instance and F is the set of instance features. In what follows, F is a set of "reduced" features, obtained by projecting measured feature vectors into a lower-dimensional space. While feature reduction is standard practice in ML [18], we show below that *aggressive* reduction allows us to efficiently manage the complexity of our game models. Behavior instances x belong to one of two classes: positive/malicious and negative/innocent (generalizing to more than two behavior classes is straightforward [18]). The goal is to learn a vector $w \in \Re^{|F|}$ such that classifier orient = $\mathrm{sign}(w^T x)$ accurately estimates the class of behavior x, returning +1 (–1) for malicious (innocent) activity.

It is useful to assess the predictability of a phenomenon before attempting to predict its evolution; for example, such an analysis permits identification of measurables that possess predictive power [20]. There has been limited theoretical work assessing predictability of adversarial dynamics, but existing studies suggest attack-defend coevolution often generates predictable dynamics. For instance, although [21] finds that certain player strategies lead to chaos in a simple repeated game, [22] shows that large sets of player strategies and repeated games exhibit predictable adversarial dynamics. Here we supplement this theoretical work by conducting an empirical investigation of predictability, and select as our case study a cyber security problem – Spam filtering – which possesses attributes that are representative of many adversarial domains.

To conduct this investigation, we first obtained a large collection of emails from various publicly-available sources for the period 1999-2006, and added to this corpus a set of Spam emails acquired from B. Guenter's Spam trap for the same time period. Following standard practice, each email is modeled as a "bag of words" feature vector $x \in \Re^{|F|}$, where the entries of x are the frequencies with which the words in vocabulary F appear in the message. The resulting dataset consists of ~128,000 emails composed of more than 250,000 features. We extracted from this collection of Spam and non-Spam emails the set of messages sent during the 30 month period between January 2001 and July 2003 (email in other periods exhibit very similar evolutionary dynamics). Finally, the dimension of the email feature space was reduced via a singular value decomposition (SVD) analysis [18], yielding a reduction in feature space dimension of four orders of magnitude (from ~250K to 20).

We wish to examine, in a simple but meaningful way, the predictability of Spam adaptation, and propose two intuitively reasonable criteria with which to empirically evaluate predictability: *sensibility* and *regularity* (obviously more comprehensive, mathematically-rigorous frameworks can be derived for defining and assessing predictability [e.g.,20]). More specifically, and in the context of Spam, it would be *sensible* for Spammers to adapt their messages over time in such a way that Spam feature vectors $x_S$ come to resemble the feature vectors $x_{NS}$ of legitimate emails, and *regularity* in this adaptation might imply that the values of the individual elements of $x_S$ approach those of $x_{NS}$ in a fairly monotonic fashion.

To permit convenient examination of the evolution of feature vectors $x_S$ and $x_{NS}$ during the 30 month period under study, the emails were first binned by quarter. Next, the average values for each of the 20 (reduced) features was computed for all the Spam emails and all the non-Spam emails (separately) for each quarter. Figure 1 illustrates the feature space dynamics of Spam and non-Spam messages for one representative element (F1) of this reduced feature space. As seen in the plot, the value of feature F1 for Spam approaches the value of this feature for non-Spam, and this increasing similarity is a consequence of changes in the composition of Spam messages (the value of F1 for non-Spam emails is essentially constant). The dynamics of the other feature values (not shown) are analogous.

Observe that the Spam dynamics illustrated in Figure 1 reflect *sensible* adaptation on the part of Spammers: the features of Spam email messages evolve to appear more like those of non-Spam email, making Spam more difficult to detect. Additionally, this evolution is *regular*, with feature values for Spam approaching those for non-Spam in a nearly-monotonic fashion. Thus this empirical analysis indicates that coevolving Spammer-Spam filter dynamics possesses some degree of predictability, and that the features employed in Spam analysis may have predictive power; this result is in general agreement with the conclusions of the theoretical predictability analysis reported in [22]. Moreover, because many of the characteristics of Spam-Spam defense coevolution are shared by other adversarial systems, this result suggests these other systems may have exploitable levels of predictability as well.

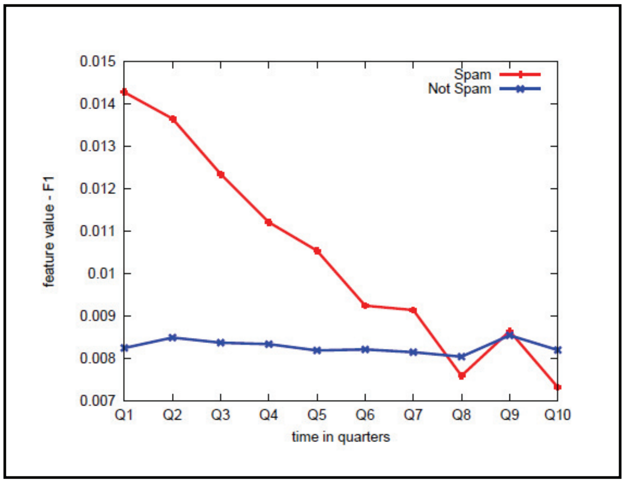


**Figure 1.** Spam/non-Spam evolution in feature space. The plot depicts evolution of feature F1 for Spam (red) and non-Spam (blue) over time (horizontal axis).

### B. *Predictive Defense Algorithm*

The proposed approach to designing a predictive defense system which works well against both current and future attacks is to combine ML with a simple game-based model for adversary behavior. In order to apply game-theoretic methods, it is necessary to overcome the complexity and model-realism challenges mentioned above. We address problem complexity by modeling adversary actions directly in an aggressively-reduced ML feature space, so that the (effective) space of possible adversary actions which must be considered is dramatically decreased. The difficulty of deriving realistic representations for attacker behavior is overcome by recognizing that the actions of attackers can be modeled as attempts to *transform* data (i.e., feature vectors x) in such a way that malicious and innocent activities are indistinguishable. (This is in contrast to trying to model the attack instances "from scratch"). It is possible to model attacker actions as transformations of data because, within an ML problem formulation, historical attack data are available in the form of training instances.

We model adversarial coevolution as a sequential game, in which the attacker and defender iteratively optimize the following objective function:

$$\min_{w} \max_{a} \left[ -\alpha\|a\|^3 + \beta\|w\|^3 + \sum_{i} \text{loss}\left(y_i, w^T(x_i + a)\right) \right] \qquad (1)$$

In (1), the loss function represents the misclassification rate for the defense system, where $\{y_i, x_i\}^n_{i=1}$ denotes pairs of currently-observed activity instances $x_i$ and their labels $y_i$ and w parameterizes the defense (recall the defense attempts to distinguish malicious and innocent activities using the classifier orient = sign($w^T x$)). The attacker attempts to circumvent the defense by transforming the data through vector $a \in \Re^{|F|}$, and the defender's goal is to optimally counter this attack through specification of the appropriate classifier vector $w \in \Re^{|F|}$. The terms $-\alpha\|a\|^3$ and $\beta\|w\|^3$ define "regularizations" imposed on attacker and defender actions, respectively, as discussed below.

Observe that (1) models the attacker as acting to increase the misclassification rate with vector a, subject to the need to limit the magnitude of this vector (large a is penalized via the term $-\alpha\|a\|^3$). This model thus captures in a simple way the fact that the actions of the attacker are in reality always constrained by the goals of the attack. For instance, in the case of Spam, the Spammer tries to manipulate message x in such a way that it "looks" enough like legitimate email to evade the Spam filter. However, the transformed message x+a must still communicate the desired information to the recipient or the attacker's goal will not be realized, and so the transformation vector a cannot be chosen arbitrarily.

The defender attempts to reduce the misclassification rate with an optimal choice for vector w, and avoids "over-fitting" through regularization with the $\beta\|w\|^3$ term [18]. Notice that the formulation (1) permits the attacker's goal to be modeled as counter to, but not exactly the opposite of, the defender's goal, and this is consistent with many real-world settings. Returning to the Spam example, the Spammer's objective of delivering messages which induce profitable user responses is not the inverse of an email service provider's goal of achieving high Spam recognition with a very low false-positive rate.

The preceding development can be summarized by stating the following predictive defense (PD) algorithm:

**Algorithm PD**

1. Collect historical data $\{y_i, x_i\}^n_{i=1}$ which reflects past behavior of the attacker and past legitimate behavior.
2. Optimize objective function (1) to obtain the predicted actions a* of the attacker and the optimal defense w* to counter this attack.
3. Estimate the status of any new activity x as either malicious (+1) or innocent (−1) via orient = sign($x^T w^*$).

Observe that Step 2 of this algorithm can be interpreted as first predicting the attacker strategy through computation of attack vector a*, and then learning an appropriate countermeasure w* by applying ML to the "transformed" data $\{y_i, x_i+a^*\}^n_{i=1}$.

### C. *Algorithm Evaluation*

This case study examines the performance of Algorithm PD for the Spam filtering problem. We use the Spam/non-Spam email dataset introduced above, consisting of ~128,000 messages that were sent during the period 1999-2006. The study compares the effectiveness of Algorithm PD, implemented as a Spam filter, with that of a well-tuned naïve Bayes (NB) Spam filter [15]. Because NB filters are widely used and work very well in Spam applications, this filter is referred to as the gold-standard algorithm. We extract from our dataset the 1000 oldest legitimate emails and 1000 oldest Spam messages for use in training both Algorithm PD and the gold-standard algorithm. The email messages sent during the four year period immediately following the date of the last training email are used as test data. More specifically, these emails are binned by quarter and then randomly sub-sampled to create balanced datasets of Spam and legitimate emails for each of the 16 quarters in the test period.

Recall that Algorithm PD employs aggressive feature space dimension reduction to manage the complexity of the game-based modeling process. This dimension reduction is accomplished here through SVD analysis, which reduces the dimension |F| of feature vectors from ~250K to 20) [18]. (The orthogonal basis used for this reduction is derived by performing SVD analysis using the 1000 non-Spam and 1000 Spam training emails.) We have found that good classification accuracy can be obtained with a wide range of (reduced) feature space dimensions. For example, we achieve a filtering accuracy of ~97% with the training data when using an NB classifier implemented with feature dimension ranging from |F|=100,000 to |F|=5.

The gold-standard strategy is applied as described in [15]. Algorithm PD is implemented with parameter values $\alpha = 0.001$ and $\beta = 0.1$, and with a sum-of-squares loss function. To evaluate the utility of these defenses against evolving adversaries, we train Algorithm PD and the gold-standard algorithm *once*, using the (oldest) 1000 non-Spam/1000 Spam dataset, and then apply the filters without retraining to the four years of emails that follow these 2000 messages.

Sample results from this study are depicted in Figure 2. Each data point in the plots represents the average accuracy over ten trials (two-fold cross-validation). It can be seen that the Spam filter based upon Algorithm PD significantly outperforms the gold-standard method: the predictive defense experiences almost no degradation in accuracy over the four years of the study, while the gold-standard method suffers a substantial drop in accuracy during this period. These results suggest that combining ML with simple game-based models offers an effective means of defending against adaptive adversaries .

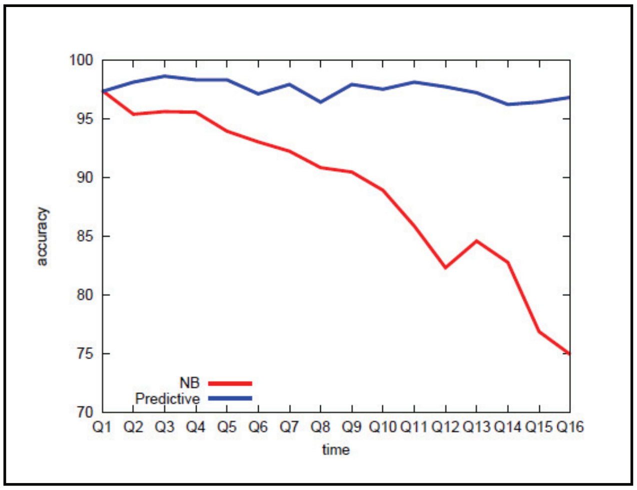


**Figure 2.** Results for the predictive defense case study. The plot shows how Spam filter accuracy (vertical axis) varies with time (horizontal axis) for the gold-standard NB filter (red) and Algorithm PD filter (blue).

### D. Randomized Feature Learning

An important consideration when applying ML techniques in adversarial settings is the extent to which adversaries can reverse-engineer the learning algorithm and use this knowledge to circumvent the classifier [3]. One way to increase the difficulty of the adversary's reverse-engineering task is to employ "randomized feature" learning [23]. Here we explore a very simple three-step implementation of this idea: 1.) divide the set of available features into randomly-selected, possibly overlapping subsets; 2.) train one classifier for each subset of features; and 3.) alternate between classifiers in a random fashion during operation. The fact that good classifier performance is often obtainable with only a few features (see the Spam example above) suggests the feasibility of employing multiple small subsets of randomly-selected features in a suite of classifiers.

To test the effectiveness of this strategy, we use a variant of the optimization process specified in (1). More specifically, we first use training data $\{y_i, x_i\}^n_{i=1}$ to computed the classifier vector w in two ways: 1.) using the full set of (reduced-dimension) features F, 2.) using two subsets of features randomly selected from set F; the resulting classifier vectors are denoted $w_F$ and $\{w_{F1}, w_{F2}\}$. (1) is then employed to compute the optimal attack against classifier vector $w_F$, denoted $a_F$, and to compute the optimal attack against the defense consisting of randomly alternating classifiers $w_{F1}$ and $w_{F2}$, denoted $a_{F12}$.

Applying this evaluation process to the 2000 email training dataset described in Section IIC suggests that randomized feature leaning may be an effective way to reduce the efficacy of adversary reverse-engineering methods. We define F to be the set of 20 features with largest singular values (in the SVD reduction process), and build sets F1 and F2 by randomly sampling F (with replacement) until each subset contains 10 features. The classification accuracy of $w_F$ against *nominal* data (i.e., with a=0) is superior to that provided by a classifier which randomly alternates between classifiers $w_{F1}$ and $w_{F2}$, but the difference is modest – the respective accuracies are 98.4% and 96.2% (two-fold cross-validation). Crucially, however, the randomized feature classifier is substantially more robust against *attack* data (i.e., data corresponding to $a=a_F$ or $a=a_{F12}$). Indeed, the accuracy of classifier $w_F$ is only 66.1% against attack data, while the accuracy of filter $\{w_{F1},w_{F2}\}$ is 86.8%, in the attack setting (two-fold cross-validation, see Figure 3).

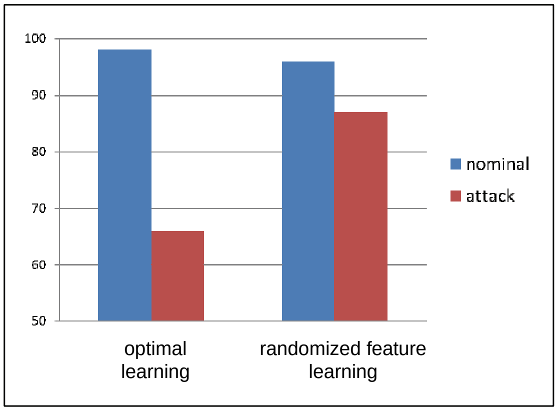


**Figure 3.** Results for randomized feature learning case study. Bar chart shows Spam/non-Spam filter accuracy for classifiers $w_F$ (left bars) and $\{w_{F1}, w_{F2}\}$ (right bars) for nominal data (blue) and "attack" data (red).

## III. Predictive Defense Via Extrapolative Learning

### *A. Problem Formulation*

The previous section derives a predictive defense system in the “feature space” of observable variables that characterize adversary activity. In this section we adopt a complementary perspective, proposing a simple technique for developing proactive defenses in “defense space”, that is, in the space of defense system parameterizations. The specific problem of interest may be stated as follows: given a (possibly limited) history of defense system configurations, design a new defense which performs well against both current and future attacks.

As noted above, it is useful to examine the predictability of a phenomenon of interest before attempting to predict its evolution [20]. Here we conduct an empirical investigation of the predictability of defense system dynamics through a case study which employs the same Spam/non-Spam email dataset introduced in Section II. The present study focuses on those messages sent during the three year period 2001-2004 (other periods exhibit very similar behavior). We assess defense system predictability in terms of the *sensibility* and *regularity* of the observed dynamics. More specifically, and in the context of Spam defense, it is *sensible* for a Spam filter to adapt to compensate for the way Spammers modify their messages over time, and in a *regular* adaptation the values of defense system parameters might change approximately monotonically.

To examine the dynamics of Spam filter configurations associated with our dataset, we first binned the messages by quarter and performed aggressive feature-space dimension reduction via SVD analysis, retaining the five features with largest singular values. Next, separate NB filters were trained for each quarter, and the filter weights {w1, w2, w3, w4, w5} corresponding to features F1-F5 were recorded. Figure 4 depicts the values of the NB filter weights for quarters 1, 5, 9, and 13 (filter weights for the other quarters are consistent with those shown in the plot and are suppressed for clarity).

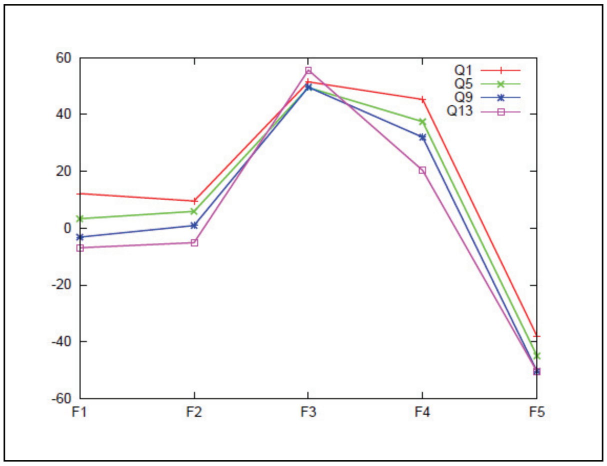


**Figure 4.** Spam filter evolution in defense space. The plot depicts values of the Spam filter weights corresponding to features F1-F5 for the first quarters of 2001 (red), 2002 (green), 2003 (blue), and 2004 (magenta).

Inspecting the evolution of filter weights depicted in Figure 4 reveals that defense adaptation is sensible. For example, by comparing Figures 1 and 4 it is seen that, as feature F1 evolves to become less predictive of Spam (Figure 1), the Spam filter places less emphasis on this feature (Figure 4); similar behavior is observed for the other weights. Additionally, the dynamics of the feature weights is regular, with most of the weights exhibiting monotonic adaptation. Thus the empirical analysis indicates that Spam filter dynamics possesses some degree of predictability, and that filter parameters may have predictive power. These results suggest the possibility that defenses in other domains may have exploitable levels of predictability as well.

### *B. Extrapolative Defense Algorithm*

The proposed approach to designing a predictive classifier that works well against both current and future attacks is to simply extrapolate the sequence of observed defense systems forward in time. Note that this strategy is motivated by the results of the empirical predictability analysis summarized above. Sequences of defense system parameterizations can often be obtained directly, for example from the system “owners”. Alternatively, if historical attack data are available, these data can be used to learn associated defense sequences (as illustrated above).

There are many ways to extrapolate a given sequence of defense system parameterizations $\{w_1, w_2, \ldots, w_p\}$ into the future, and thereby generate predictions for useful future defenses. We adopt the following linear strategy:

$$w_{p+T} = \Sigma^{p}_{i=1} \beta_i w_i \tag{2}$$

where the $w_i$ and $\beta_i$ are defense parameterizations and extrapolation coefficients, respectively, and T is the time horizon for which a prediction is desired. The coefficients $\beta_i$ are ordinarily specified so that $|\beta_i| \geq |\beta_j|$ if $i > j$, so more recent observations are emphasized. Appropriate values for the $\beta_i$ can be estimated in various ways, including statistical inference from historical data [18] or consultation with domain experts [15].

The preceding discussion can be summarize by sketching an algorithm for predicting a classifier vector $w_{p+T}$ which may be expected to be useful at future time $t=p+T$:

**Algorithm ED (Extrapolative Defense)**

1. Collect a sequence of defense system parameterizations $\{w_1, w_2, \ldots, w_p\}$ (e.g., from historical defense data or by learning appropriate defenses from historical attack data).
2. Estimate the coefficients $\beta_i$ in (2) (e.g., using ML).
3. Compute classifier vector $w_{p+T}$ from (2), and estimate the status of any new activity as either malicious (+1) or innocent (−1) via orient = $\mathrm{sign}(x^T w_{p+T})$.

### *C. Algorithm Evaluation*

This case study examines the performance of Algorithm ED for the Spam filtering problem. We use the Spam/non-Spam email dataset described above, consisting of all emails sent during the 54 month period from early 2001 to mid-2005. The study compares the effectiveness of Algorithm ED, implemented as a Spam filter, with that of two NB Spam filters trained in different ways. As in the previous case studies, we first binned the emails by quarter, and then randomly sampled each quarter to create balanced datasets for all 18 quarters in the study period.

To provide a demanding test, we extracted from our dataset the emails sent during quarters Q1, Q5, and Q9 for use in training Algorithm ED. This procedure is intended to reflect the common situation in which opportunities for observation may arise only sporadically. The messages sent during the 18 month period from quarters Q13 to Q18 serve as test data. (The quarters closest to the training period, Q10 through Q12, are not included in the test set to increase the difficulty of the task.)

Algorithm ED is implemented by first training NB filters on data from quarters Q1, Q5, and Q9, yielding defense parameterizations $\{w_1, w_5, w_9\}$, and then using (2) to extrapolate these defenses. More specifically, we compute predicted defense $w^*$ using (2) with $\beta_1 = 0$, $\beta_5 = -1$, and $\beta_9 = 2$ (a simple Euler-like extrapolation). The first NB filter used for comparison employs $w_9$, that is, the filter derived from the most recent training data. The second NB filter examined in this case study is permitted to use "future" data during training: when attempting to distinguish Spam and non-Spam emails in quarter Qm, for $m \in \{13, 14, \ldots, 18\}$, this filter is trained on Qm data. Because the latter NB filter has access to future data, which is unavailable to the other defense systems, the performance of this filter is expected to be an upper bound for that of a predictive filter, and we refer to this NB filter as the gold-standard. All three filters – Algorithm ED, nominal NB, and gold-standard – are applied using an aggressively-reduced feature space of dimension $|F|=5$.

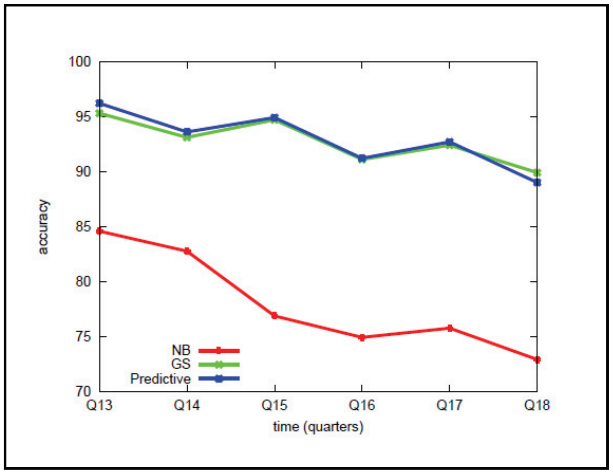


**Figure 5.** Results for the extrapolative defense case study. The plot shows how Spam filter accuracy (vertical axis) varies with time (horizontal axis) for the nominal NB filter, (red), gold-standard NB filter (green), and Algorithm ED filter (blue).

Sample results from this study are shown in Figure 5. Each data point in the plots represents the average accuracy over ten trials (two-fold cross-validation). It is seen that the filter based upon Algorithm ED significantly outperforms the nominal NB method. Moreover, the accuracy of Algorithm ED is comparable to that achieved by the gold-standard NB method, despite the fact that the latter filter is trained on "future" data not available to Algorithm ED. These results suggest that simple defense system extrapolation offers an effective means of defending against evolving adversary behavior.

## Acknowledgements

This work was supported by the Laboratory Directed Research and Development Program at Sandia National Laboratories. We thank Chip Willard and Curtis Johnson for numerous helpful discussions on aspects of this research.

## References

[1] *Proc. 2010 IEEE ISI*, Vancouver, BC Canada, May 2010.

[2] *Proc. 2011 IEEE ISI*, Beijing, China, July 2011.

[3] "Machine learning in adversarial environments", P. Laskov, R. Lippmann, Eds, Special Issue, *Machine Learning*, Vol. 81, 2010.

[4] Zhang, Q., D. Man, and W. Yang, "Using HMM for intent recognition in cyber security situational awareness", *Proc. IEEE KAM*, Wuhan, China, November 2009.

[5] Parameswaran, M., H. Rui, and S. Sayin, "A game theoretic model and empirical analysis of Spammer strategies", *Proc. CEAS 2010*, Redmond, WA, July 2010.

[6] Ahmadinejad, S., S. Jalili, and M. Abadi, "A hybrid model for correlating alerts of known and unknown attack scenarios and updating attack graphs", *Computer Networks*, Vol. 55, pp. 2221-2240, 2011.

[7] Zakrzewska, A. and E. Ferragut, "Modeling cyber conflicts using an extended Petri Net formalism", *Proc. IEEE CICS*, Paris, France, April 2011.

[8] Kaza, S., Y. Wang, and H. Chen, "Enhancing border security: Mutual information analysis to identify suspect vehicles", *Decision Support Systems*, Vol. 43, pp. 199-210, 2007.

[9] Gkonis, K. and H. Psaraftis, "Container transportation as an interdependent security problem", *J. Transportation Security*, Vol. 3, pp. 197-211, 2010.

[10] Pita, J. et al., "GUARDS: Game theoretic security allocation on a national scale", *Proc. AAMAS '11*, Taipei, Taiwan, May 2011.

[11] Williams, E., *Surveillance and Interdiction Models: A Game Theoretic Approach to Defend Against VBIED*, Thesis, Naval Postgraduate School, June 2010.

[12] Smith, A., "Improvised explosive devices in Iraq, 2003-09", *The Letort Papers*, US Army War College, April 2011.

[13] Colbaugh, R., "Does coevolution in malware adaptation enable predictive analysis?", *IFA Workshop: Exploring Malware Adaptation Patterns*, San Francisco, CA, May 2010.

[14] Bozorgi, M., L. Saul, S. Savage, and G. Voelker, "Beyond heuristics: Learning to classify vulnerabilities and predict exploits", *Proc. ACM KDD '10*, Washington DC, July 2010.

[15] Colbaugh, R. and K. Glass, "Proactive defense for evolving cyber threats", *Proc. 2011 IEEE ISI*, Beijing, China, July 2011.

[16] Cipriano, C. et al., "NEXAT: History-based approach to predict attacker actions", *Proc. ACSAC*, Orlando, FL, December 2011.

[17] Peters, H., *Game Theory*, Springer, Berlin, 2008.

[18] Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Second Edition, Springer, New York, 2009.

[19] Dalvi, N. et al., "Adversarial classification", *Proc. ACM KDD '09*, Seattle, WA, August 2004.

[20] Colbaugh, R. and K. Glass, "Predictive analysis for social processes I: Multi-scale hybrid system modeling, and II: Predictability and warning analysis", *Proc. 2009 IEEE MSC*, Saint Petersburg, Russia, July 2009.

[21] Sato, Y., E. Akiyama, and J.D. Farmer, "Chaos in learning a simple two-person game", *Proc. National Academy of Sciences USA*, Vol. 99, pp. 4748-4751, 2002.

[22] Colbaugh, R., "Arctic ice, George Clooney, lipstick on a pig, and insomniac fruit flies: Combining kd and m&s for predictive analysis", *Proc. ACM KDD '11*, San Diego, CA, August 2011.

[23] Johnson, C., Personal communication, December 2011.

# Proactive Defense for Evolving Cyber Threats

Richard Colbaugh
Sandia National Laboratories
New Mexico Institute of Mining and Technology
Albuquerque, NM USA
colbaugh@comcast.net

Kristin Glass
New Mexico Institute of Mining and Technology
Socorro, NM USA
kglass@icasa.nmt.edu

*Abstract*—**There is significant interest to develop proactive approaches to cyber defense, in which future attack strategies are anticipated and these insights are incorporated into defense designs. This paper considers the problem of protecting computer networks against intrusions and other attacks, and leverages the coevolutionary relationship between attackers and defenders to derive two new methods for proactive network defense. The first method is a bipartite graph-based machine learning algorithm which enables information concerning previous attacks to be "transferred" for application against novel attacks, thereby substantially increasing the rate with which defense systems can successfully respond to new attacks. The second approach involves exploiting basic threat information (e.g., from cyber security analysts) to generate "synthetic" attack data for use in training defense systems, resulting in networks defenses that are effective against both current and (near) future attacks. The utility of the proposed methods is demonstrated by showing that they outperform standard techniques for the task of detecting malicious network activity in two publicly-available cyber datasets.**

***Keywords***—-cyber security, proactive defense, predictive analysis, machine learning, security informatics.

## I. Introduction

Rapidly advancing technologies and evolving operational practices and requirements increasingly drive both private and public sector enterprises toward highly interconnected and technologically convergent information networks. Proprietary information processing solutions and stove-piped databases are giving way to unified, integrated systems, thereby dramatically increasing the potential impact of even a single well-planned network intrusion, data theft, or denial-of-service attack. It is therefore essential that commercial and government organizations develop network defenses which are able to respond rapidly to, or even foresee, new attack strategies and tactics.

Recognizing these trends and challenges, some cyber security researchers and practitioners are focusing their efforts on developing *proactive* methods of cyber defense, in which future attack strategies are anticipated and these insights are incorporated into defense designs [e.g., 1-5]. However, despite this attention, much remains to be done to place the objective of proactive defense on a rigorous and quantitative foundation. Fundamental issues associated with the dynamics and predictability of the coevolutionary "arms race" between attackers and defenders have yet to be resolved. For instance, although recent work has demonstrated that previous attacker actions and defender responses provide predictive information about future attacker behavior [3-5], not much is known about which measurables have predictive power or how to exploit these to form useful predictions. Moreover, even if these predictability and prediction issues were resolved, it is still an open question how to incorporate such predictive analytics into the design of practically-useful cyber defense systems.

This paper considers the problem of protecting enterprise-scale computer networks against intrusions and other attacks, and explicitly leverages the coevolutionary relationship between attackers and defenders to develop two new methods for proactive network defense. Each method formulates the task as one of behavior classification, in which innocent and malicious network activities are to be distinguished, and each assumes that only very limited prior information is available regarding exemplar attacks or attack attributes. The first method models the data as a bipartite graph of *instances* of network activities and the *features* or attributes that characterize these instances. The bipartite graph data model is used to derive a machine learning algorithm which accurately classifies a given instance as either innocent or malicious based upon its behavioral features. The algorithm enables information concerning previous attacks to be "transferred" for use against novel attacks; crucially, it is assumed that previous attacks are drawn from a distribution of attack instances which is related *but not identical* to that associated with the new malicious behaviors. This transfer learning algorithm provides a simple, effective way to extrapolate attacker behavior into the future, and thus significantly increases the rate with which defense systems can successfully respond to new attacks.

The second approach to proactive network defense proposed in this paper represents attacker-defender coevolution as a hybrid dynamical system (HDS) [6,7], with the HDS discrete system modeling the "modes" of attack (e.g., a particular class of DoS or data exfiltration procedures) and the HDS continuous system generating particular attack instances corresponding to the attack mode presently "active". Our algorithm takes as input the mode of attack, obtained for example from the insights of cyber analysts, and generates synthetic attack data for this mode of malicious activity; these data are then combined with actually observed attacks to train a learning-based classifier to be effective against both current and (near) future attacks. The utility of the proposed methods is demonstrated by showing that they outperform standard techniques for the task of distinguishing innocent and malicious network behaviors in analyses of two publicly-available cyber datasets.

## II. Preliminaries

We approach the task of protecting computer networks from attack as a classification problem, in which the objective is to distinguish innocent and malicious network activity. Each instance of network activity is represented as a feature vector $x \in \Re^{|F|}$, where entry $x_i$ of x is the value of feature i for instance x and F is the set of instance features or attributes of interest (x may be normalized in various ways [7]). Instances can belong to one of two classes: positive / innocent and negative / malicious; generalizing to more than two classes is straightforward. We wish to learn a vector $c \in \Re^{|F|}$ such that the classifier orient = sign($c^T x$) accurately estimates the class label of behavior x, returning +1 (−1) for innocent (malicious) activity.

Knowledge-based classifiers leverage prior domain information to construct the vector c. One way to obtain such a classifier is to assemble a “lexicon” of positive / innocent features $F^+ \subseteq F$ and malicious / negative features $F^- \subseteq F$, and to set $c_i = +1$ if feature i belongs to $F^+$, $c_i = -1$ if i is in $F^-$, and $c_i = 0$ otherwise; this classifier simply sums the positive and negative feature values in the instance and assigns instance class accordingly. Unfortunately this sort of scheme is unable to improve its performance or adapt to new domains, and consequently is usually not very useful in cyber security applications.

Alternatively, learning-based methods attempt to generate the classifier vector c from examples of positive and negative network activity. To obtain a learning-based classifier, one can begin by assembling a set of $n_l$ *labeled* instances $\{(x_i, d_i)\}$, where $d_i \in \{+1, -1\}$ is the class label for instance i. The vector c is then learned through training with the set $\{(x_i, d_i)\}$, for example by solving the following set of equations for c:

$$[X^T X + \gamma I_{|F|}]\, c = X^T d, \qquad (1)$$

where matrix $X \in \Re^{n_l \times |F|}$ has instance feature vectors for rows, $d \in \Re^{n_l}$ is the vector of instance labels, $I_{|F|}$ denotes the $|F| \times |F|$ identity matrix, and $\gamma \geq 0$ is a constant; this corresponds to regularized least squares (RLS) learning [8]. Many other learning strategies can be used to compute c [8]. Learning-based classifiers have the potential to improve their performance and adapt to new situations, but realizing these capabilities typically requires that large training sets of labeled attacks be obtained. This latter characteristic represents a significant drawback for cyber security applications, where it is desirable to be able to recognize new attacks given only a few (or no) examples.

In what follows we present two new learning-based approaches to cyber defense which are able to perform well with only very modest levels of prior knowledge regarding the attack classes of interest. The basic idea is to leverage “auxiliary” information which is readily available in cyber security applications. More specifically, the first proposed method is a *transfer learning* algorithm [e.g., 9] which permits the knowledge present in data on previous attacks to be transferred for implementation against new attacks. The second approach uses prior knowledge concerning attack “modes” to generate synthetic attack data for use in training defense systems, resulting in networks defenses which are effective against both current and (near) future attacks.

## III. Method One: Transfer Learning

In this section we first derive a bipartite graph-based transfer learning algorithm for distinguishing innocent and malicious network behaviors, and then demonstrate the algorithm’s effectiveness through a case study using publicly-available network intrusion data obtained from the KDD Cup archive [10]. The basic hypothesis is simple and natural: because attacker / defender behavior coevolves, previous activity should provide some indication of future behavior, and transfer learning is one way to quantify and operationalizes this intuition.

### A. *Proposed Algorithm*

The development of the proposed algorithm begins by modeling the problem data as a bipartite graph $G_b$, in which instances of network activity are connected to their features (see Figure 1). It is easy to see that the adjacency matrix A for graph $G_b$ is given by

$$A = \begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix} \qquad (2)$$

where matrix $X \in \Re^{n \times |F|}$ is constructed by stacking the n instance feature vectors as rows, and each ‘0’ is a matrix of zeros. In the proposed algorithm, integration of labeled and “auxiliary” data is accomplished by exploiting the relationships between instances and features encoded in the bipartite graph model. The basic idea is to assume that, in $G_b$, positive / negative instances will tend to be connected to positive / negative features. Note that, as shown below, the learning algorithm can incorporate a lexicon of labeled features (if available). It is assumed that this lexicon is used to build vector $w \in \Re^{|F|}$, where the entries of w are set to +1 (innocent), −1 (malicious), or 0 (unknown) according to the polarity of the corresponding features.

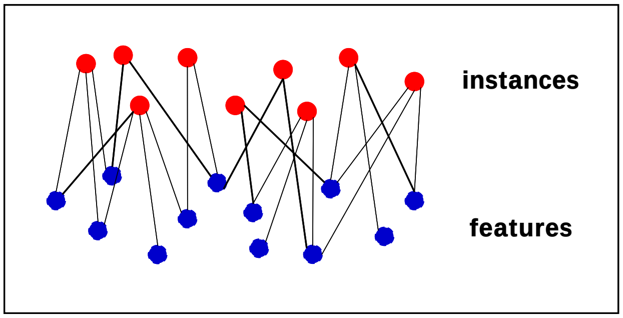


**Figure 1.** Cartoon of bipartite graph model $G_b$, in which the instances of network activity (red vertices) are connected to features (blue vertices) they contain, and link weights (black edges) reflect the magnitudes taken by the features in the associated instances.

Many cyber security applications are characterized by the presence of limited labeled data for the attack class of interest but ample labeled information for a related class of malicious activity. For example, an analyst may be interested in detecting a new class of attacks, and may have in hand a large set of labeled examples of normal network behavior as well as attacks which have been experienced in the recent past. In this setting it is natural to adopt a transfer learning approach, in which knowledge concerning previously observed instances of innocent / malicious behavior, the so-called *source* data, is transferred to permit classification of new *target* data. In what follows we present a new bipartite graph-based approach to transfer learning that is well-suited to cyber defense applications.

Assume that the initial problem data consists of a collection of n = $n_T + n_S$ network events, where $n_T$ is the (small) number of labeled instances available for the target domain, that is, examples of network activity of current interest, and $n_S >> n_T$ is the number of labeled instances from some related source domain, say reflecting recent activity; suppose also that a modest lexicon $F_l$ of labeled features is known (this set can be empty). Let this label data be used to encode vectors $d_T \in \Re^{nT}$, $d_S \in \Re^{nS}$, and $w \in \Re^{|F|}$, respectively. Denote by $d_{T,est} \in \Re^{nT}$, $d_{S,est} \in \Re^{nS}$, and $c \in \Re^{|F|}$ the vectors of estimated class labels for the target and source instances and the features, and define the *augmented classifier* $c_{aug} = [d_{S,est}^T \;\; d_{T,est}^T \;\; c^T]^T \in \Re^{n+|F|}$. Note that the quantity $c_{aug}$ is introduced for notational convenience in the subsequent development and is not directly employed for classification.

We derive an algorithm for learning $c_{aug}$, and therefore c, by solving an optimization problem involving the labeled source and target training data, and then use c to estimate the class label of any new instance of network activity via the simple linear classifier orient = sign($c^Tx$). This classifier is referred to as *transfer learning-based* because c is learned, in part, by transferring knowledge about the way innocent and malicious network behavior is manifested in a domain which is related to (but need not be identical to) the domain of interest.

We wish to learn an augmented classifier $c_{aug}$ with the following four properties: 1.) if a source instance is labeled, then the corresponding entry of $d_{S,est}$ should be close to this ±1 label; 2.) if a target instance is labeled, then the corresponding entry of $d_{T,est}$ should be close to this ±1 label, and the information encoded in $d_T$ should be emphasized relative to that in the source labels $d_{S,}$; 3.) if a feature is in the lexicon $F_l$, then the corresponding entry of c should be close to this ±1 label; and 4.) if there is an edge $X_{ij}$ of $G_b$ which connects an instance i and a feature j, and $X_{ij}$ possesses significant weight, then the estimated class labels for i and j should be similar.

The four objectives listed above may be realized by solving the following minimization problem:

$$\min_{c_{aug}} \; c_{aug}^T L c_{aug} + \beta_1 \left\| d_{S,est} - k_S d_S \right\|^2 + \beta_2 \left\| d_{T,est} - k_T d_T \right\|^2 + \beta_3 \left\| c - w \right\|^2 \tag{3}$$

where L = D − A is the graph Laplacian matrix for $G_b$, with D the diagonal degree matrix for A (i.e., $D_{ii} = \Sigma_j A_{ij}$), and $\beta_1$, $\beta_2$, $\beta_3$, $k_S$, and $k_T$ are nonnegative constants. Minimizing (3) enforces the four properties we seek for $c_{aug}$. More specifically, the second, third, and fourth terms penalize "errors" in the first three properties, and choosing $\beta_2 > \beta_1$ and $k_T > k_S$ favors target label data over source labels. To see that the first term enforces the fourth property, note that this expression is a sum of components of the form $X_{ij}$ $(d_{T,est,i} - c_j)^2$ and $X_{ij}$ $(d_{S,est,i} - c_j)^2$. The constants $\beta_1$, $\beta_2$, $\beta_3$ can be used to balance the relative importance of the four properties.

The $c_{aug}$ which minimizes the objective function (3) can be obtained by solving the following set of linear equations:

$$\begin{bmatrix} L_{11} + \beta_1 I_{nS} & L_{12} & L_{13} \\ L_{21} & L_{22} + \beta_2 I_{nT} & L_{23} \\ L_{31} & L_{32} & L_{33} + \beta_3 I_{|F|} \end{bmatrix} c_{aug} = \begin{bmatrix} \beta_1 k_S d_S \\ \beta_2 k_T d_T \\ \beta_3 w \end{bmatrix} \tag{4}$$

where the $L_{ij}$ are matrix blocks of L of appropriate dimension. The system (4) is sparse because the data matrix X is sparse, and therefore large-scale problems can be solved efficiently. Note that in situations where the set of available labeled instances and features is *very* limited, classifier performance can be improved by replacing L in (4) with the normalized Laplacian $L_n = D^{-1/2} L D^{-1/2}$, or with a power of this matrix $L_n^k$ (for k a positive integer).

We summarize the above discussion by sketching an algorithm for constructing the proposed transfer learning classifier:

**Algorithm TL (Transfer Learning):**

1. Assemble the set of equations (4), possibly by replacing the graph Laplacian L with $L_n^k$.
2. Solve equations (4) for $c_{aug} = [d_{S,est}^T \;\; d_{T,est}^T \;\; c^T]^T$ (for instance using the Conjugate Gradient method).
3. Estimate the class label (innocent or malicious) of any new activity x of interest as: orient = sign($c^Tx$).

### *B. Algorithm Evaluation*

We now examine the performance of Algorithm TL for the problem of distinguishing innocent and malicious network activity in the KDD Cup 99 dataset, a publicly-available collection of network data consisting of both normal activities and attacks of various kinds [10]. For this study we randomly selected 1000 Normal connections (N), 1000 denial-of-service attacks (DoS), and 1000 unauthorized remote-access events (R2L) to serve as our test data. Additionally, small sets of each of these classes of activity were chosen at random from [10] to be used for training Algorithm TL, and a lexicon of four features, two positive and two negative, was constructed manually and employed to form the lexicon vector w.

We defined two tasks with which to explore the utility of Algorithm TL. In the first, the goal is to distinguish N and DoS instances, and it is assumed that the following data is available to train Algorithm TL: 1.) a set of $d_S/2$ labeled N and $d_S/2$ labeled R2L instances (source data), 2.) a set of $d_T/2$ labeled N and $d_T/2$ labeled DoS instances (target data), and 3.) the four lexicon features. Thus the source domain consists of N and R2L activities and the target domain is composed of N and DoS instances. In the second task the situation is reversed – the objective is to distinguish N and R2L activities, the source domain is made up of $d_S$ (total) labeled N and DoS instances, and the target domain consists of $d_T$ (total) N and R2L instances. In all tests the number of labeled source instances is $d_S = 50$, while the number of target instances $d_T$ is varied to explore the way classifier performance depends on this key parameter. Of particular interest is determining if it is possible to obtain good performance with only limited target data, as this outcome would suggest both that useful information concerning a given attack class is present in other attacks *and* that Algorithm TL is able to extract this information.

This study compared the classification accuracy of Algorithm TL with that of a well-tuned version of the RLS algorithm (1) and a standard naïve Bayes (NB) algorithm [11]; as the performance of the RLS and NB methods were quite similar, we report only the RLS results. Algorithm TL is implemented with the following parameter values: $\beta_1 = 1.0$, $\beta_2 = 3.0$, $\beta_3 = 5.0$, $k_S = 0.5$, $k_T = 1.0$, and $k = 5$. We examined training sets which incorporated the following numbers of target instances: $n_T = 2, 5, 10, 20, 30, 40, 50, 60$. As in previous studies (see, for example, [10]), only the 34 "continuous features" were used for learning the classifiers.

Sample results from this study are depicted in Figure 2. Each data point in the plots represents the average of 100 trials. It can be seen that Algorithm TL outperforms the RLS classifier (and also the standard NB algorithm), and that the difference in accuracy of the methods increases substantially as the volume of training data from the target domain becomes small. The performance of Algorithm TL for this task is also superior to that reported for other learning methods tested on these data [e.g., 12]. The ability of Algorithm TL to accurately identify a novel attack after seeing only a very few examples of it, which is a direct consequence of its ability to transfer useful knowledge from related data, is expected to be of considerable value for a range of cyber security applications.

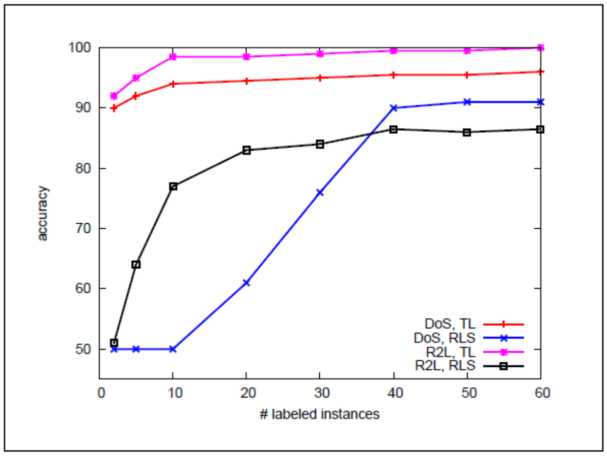


**Figure 2**. Performance of Algorithm TL with limited labeled data. The plot shows how classifier accuracy (vertical axis) varies with number of available labeled target instances (horizontal axis) for four tasks: distinguish N and DoS using RLS classifier (blue), distinguish N and DoS using Algorithm TL (red), distinguish N and R2L using RLS classifier (black), and distinguish N and R2L using Algorithm TL (magenta),

Finally, it is interesting to observe that the bipartite graph formulation of Algorithm TL permits useful information to be extracted from network data *even if no labeled instances are available*. More specifically, we repeated the above study for the case in which $d_T = d_S = 0$, that is, when no labeled instances are available in either the target or source domains. The knowledge reflected in the lexicon vector w is still made available to Algorithm TL. As shown in Figure 3, employing a "lexicon only" classifier, as described in Section II, yields classification accuracy which is not much better than the 50% baseline achievable with random guessing. However, using this lexicon information together with Algorithm TL enables useful classification accuracy to be obtained (see Figure 3). This somewhat surprising result can be explained as follows: the "clustering" property of Algorithm TL encoded in objective function (3) allows the domain knowledge in the lexicon to leverage latent information present in the *unlabeled* target and source instances, thereby boosting classifier accuracy.

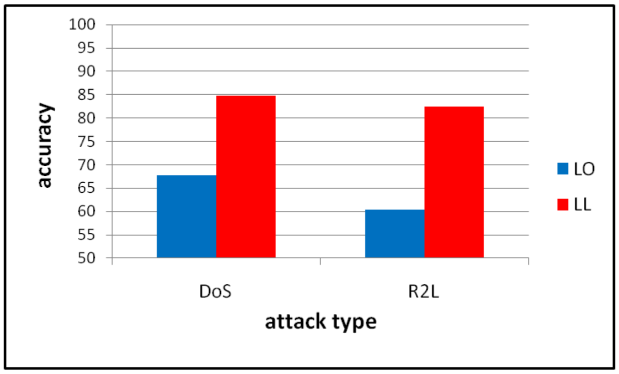


**Figure 3**. Performance of Algorithm TL with no labeled instance data. The bar graphs depicts classifier accuracy for four tasks: distinguish N and DoS using a lexicon-only (LO) classifier (left, blue bar), distinguish N and DoS using lexicon-learning (LL) via Algorithm TL (left, red bar), distinguish N and R2L using an LO classifier (right, blue bar), and distinguish N and R2L using LL via Algorithm TL (right, red bar).

## IV. Method Two: Synthetic Attack Generation

In this section we derive our second algorithm for distinguishing normal and malicious network activity and demonstrate its effectiveness through a case study using the publicly-available Ling-Spam dataset [13]. Again the intuition is that attacker / defender coevolution should make previous activity somewhat indicative of future behavior, and in the present case we exploit this notion by generating "predicted" attack data and using this synthetic data for classifier training.

### A. *Proposed Algorithm*

The development of the second approach to proactive defense begins by modeling attacker / defender interaction as a stochastic hybrid dynamical system (S-HDS). Here we present a brief, intuitive overview of the basic idea; a comprehensive description of the modeling procedure is detailed in [7]. An S-HDS (see Figure 4) is a feedback interconnection of a discrete-state stochastic process, such as a Markov chain, with a family of continuous-state stochastic dynamical systems [6,14]. Combining discrete and continuous dynamics within a unified, computationally tractable framework offers an expressive, scalable modeling environment that is amenable to formal mathematical analysis. In particular, S-HDS models can be used to efficiently represent dynamical phenomena which evolve on a broad range of time scales, a property of considerable value in the present application [14].

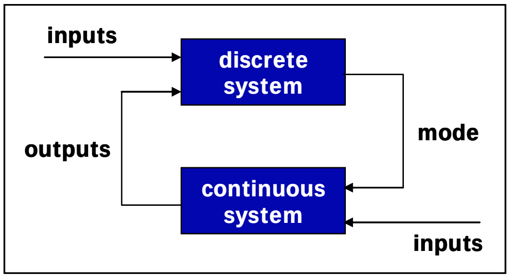


**Figure 4**. Schematic of basic S-HDS feedback structure. The discrete and continuous systems in this framework model the selection of attack "mode" and resulting adversary behavior, respectively, which arise from the coevolving attacker-defender dynamics.

As a simple illustration of the way the S-HDS formalism enables effective, efficient mathematical representation of cyber phenomena, consider the task of modeling the coevolution of Spam attack methods and Spam filters. At an abstract but still useful level, one can think of Spam-Spam filter dynamics as evolving on two timescales:

- the *slow timescale*, which captures the evolution of attack strategies; as an example, consider the way early Spam filters learned to detect Spam by identifying words that were consistently associated with Spam, and how Spammers responded by systematically modifying the wording of their messages, for instance via "add-word" (AW) and "synonym" attacks [15];
- the *fast timescale*, which corresponds to the generation of particular attack instances for a given "mode" of attack (for example, the synthesis of Spam messages according to a specific AW attack method).

We show in [7] that a range of adversarial behavior can be represented within the S-HDS framework, and derive simple but reasonable models for Spam-Spam filter dynamics and for basic classes of network intrusion attacks.

In [14] we develop a mathematically-rigorous procedure for predictive analysis for general classes of S-HDS. Among other capabilities, this analytic methodology enables the predictability of a given dynamics to be assessed and the predictive measurables (if any) to be identified. Applying this predictability assessment process to the adversarial S-HDS models constructed in [7] reveals that, for many such systems, the most predictive measurable is the *mode* of attack, that is, the state variable for the discrete system component of the S-HDS (see [7] for a detailed description of this analysis). Observe that this result is intuitively sensible.

This analytic finding suggests the following *synthetic data learning* (SDL) approach to proactive defense. First, identify the mode(s) of attack of interest. For attacks which are already underway, [7] offers an S-HDS discrete-system state estimation method that allows the mode to inferred using only modest amounts of measured data. Alternatively, and of more interest in the present application, it is often possible to identify likely future attack modes through analysis of auxiliary information sources (e.g., the subject matter knowledge possessed by domain experts or "non-cyber" data such as that found in social media [16,17]).

Once a candidate attack mode has been identified, synthetic attack data corresponding to the mode can be generated by employing one of the S-HDS models derived in [7]. The synthetic data take the form of a set of K network attack instance vectors, denoted $A_S = \{x_{S1}, \ldots, x_{SK}\}$. The set $A_S$ can then be combined with (actual) measurements of L normal network activity instances, $N_M = \{x_{NM1}, \ldots, x_{NML}\}$, and P (recently) observed attacks, $A_M = \{x_{M1}, \ldots, x_{MP}\}$, yielding the training dataset $TR = N_M \cup A_M \cup A_S$ of real and synthetic data. It is hypothesized that training classifiers with the augmented set TR may offer a mechanism for deriving defenses which are effective against both current and near future malicious activity.

We summarize the above discussion by sketching a procedure for constructing the new SDL classifier:

**Algorithm SDL:**

1. Identify the mode(s) of attack of interest (e.g., via domain experts or auxiliary data).
2. Generate a set of synthetic attack instances $A_S$ corresponding to the attack mode identified in Step 1.
3. Assemble sets of normal network activity N and measured attack activity $A_M$ for the network under study.
4. Train a classifier (e.g., RLS, NB) using the training data $TR = N_M \cup A_M \cup A_S$. Estimate the class label (innocent or malicious) of any network activity x with the formula: $\text{orient}(x) = \text{sign}(c^T x)$.

### B. *Algorithm Evaluation*

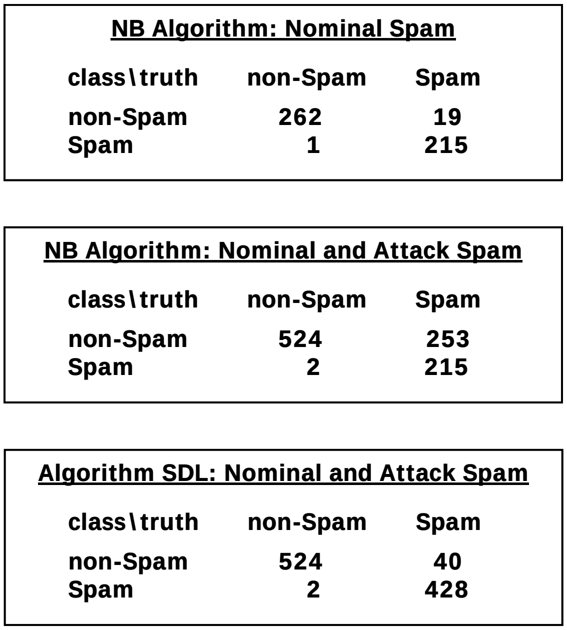
We now examine the performance of Algorithm SDL for the problem of distinguishing legitimate and Spam emails in the Ling-Spam dataset [13], a corpus of 2412 non-Spam emails collected from a linguistics mailing list and 481 Spam emails received by the list. After data cleaning and random subsampling of the non-Spam messages we are left with 468 Spam and 526 non-Spam messages for training and testing purposes; this set of 994 emails will be referred to as the *nominal Spam* corpus. (Note that all email was preprocessed using the *ifile* tool [18].)

We considered three scenarios in this study:

1. NB classifier / nominal Spam: for each of ten runs, the nominal Spam corpus was randomly divided into equal-sized training and testing sets and the class label for each message in the test set was estimated with a trained naïve Bayes (NB) algorithm [11];

2. NB classifier / nominal plus attack Spam: for each of ten runs, the nominal Spam corpus was randomly divided into equal-sized training and testing sets and the test set was then augmented with 263 additional non-Spam messages (taken from the Ling-Spam dataset) and 234 Spam messages generated via a standard add-word (AW) attack methodology [15]; the class labels for the test messages were estimated with the NB algorithm [11] trained on the nominal Spam training set;
3. Algorithm SDL / nominal plus attack Spam: for each of ten runs, the training and test corpora were constructed exactly as in Scenario 2 and the class labels for the test messages were estimated with Algorithm SDL.

In generating the AW attacks in Scenarios 2. and 3., we assume that the attacker knows to construct AW Spam to defeat an NB filter but does not have knowledge of the specific filter involved [15]. Analogously, the synthetic AW attacks generated in Scenario 3 (using Step 2 of Algorithm SDL) are computed with no knowledge of the attacker's methodology beyond the mode of attack (i.e., AW).



**NB Algorithm: Nominal Spam**

| class\truth | non-Spam | Spam |
|---|---|---|
| non-Spam | 262 | 19 |
| Spam | 1 | 215 |

**NB Algorithm: Nominal and Attack Spam**

| class\truth | non-Spam | Spam |
|---|---|---|
| non-Spam | 524 | 253 |
| Spam | 2 | 215 |

**Algorithm SDL: Nominal and Attack Spam**

| class\truth | non-Spam | Spam |
|---|---|---|
| non-Spam | 524 | 40 |
| Spam | 2 | 428 |

**Figure 5**. Performance of Algorithm SDL on Spam dataset. Each confusion matrix shows number of non-Spam messages classified as non-Spam and Spam (left column) and number of Spam messages classified as non-Spam and Spam (right column). The three matrices, from top to bottom, report the results for: NB against nominal Spam, NB against Spam which contains add-word attacks, and Algorithm SDL against Spam which contains add-word attacks.

Sample results from this study are displayed in Figure 5. In each case the "confusion matrix" [8] reports the (rounded) average performance over the ten runs. It can be seen that, as expected, the NB filter does well against the nominal Spam but poorly against the AW Spam (in fact, the NB filter does not detect a single instance of AW Spam). In contrast, Algorithm SDL performs well against both nominal Spam and AW Spam, achieving ~96% classification accuracy with a low false positive rate. It is emphasized that this result is obtained using only the (synthetic) estimate of AW Spam generated in Step 2 of Algorithm SDL.

## ACKNOWLEDGEMENTS

This work was supported by the Laboratory Directed Research and Development Program at Sandia National Laboratories. We thank Chip Willard of the U.S. Department of Defense for numerous helpful discussions on aspects of this research.

## REFERENCES

[1] Byers, S. and S. Yang, "Real-time fusion and projection of network intrusion activity", *Proc. ISIF/IEEE Intern. Conference on Information Fusion*, Cologne, Germany, July 2008.

[2] Armstrong, R., J. Mayo, and F. Siebenlist, "Complexity science challenges in cybersecurity", Sandia National Laboratories SAND Report, March 2009.

[3] Colbaugh, R., "Does coevolution in malware adaptation enable predictive analysis?", *IFA Workshop: Exploring Malware Adaptation Patterns*, San Francisco, CA, May 2010.

[4] Mashevsky, Y., Y. Namestnikov, N. Denishchenko, and P. Zelensky, "Method and system for detection and prediction of computer virus-related epidemics", US Patent 7,743,419, June 2010.

[5] Bozorgi, M., L. Saul, S. Savage, and G. Voelker, "Beyond heuristics: Learning to classify vulnerabilities and predict exploits", *Proc. ACM SIGKDD Conference*, Washington DC, July 2010.

[6] Majumdar, R. and P. Tabuada, *Hybrid Systems: Computation and Control*, LNCS 5469, Springer, Berlin, 2009.

[7] Colbaugh, R. and K. Glass, "Proactive defense for evolving cyber threats", Sandia National Laboratories SAND Report, March 2011.

[8] Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Second Edition, Springer, New York, 2009.

[9] Pan, S. and Q. Yang, "A survey on transfer learning", *IEEE Trans. Knowledge and Data Engineering*, Vol. 22, pp. 1345-1359, 2010.

[10] http://kdd.ics.uci.edu/databases/kddcup99/; accessed Dec. 2010.

[11] http://www.borgelt.net/bayes.html; accessed July 2010.

[12] He, J., Y. Liu, and R. Lawrence, "Graph-based transfer learning", *Proc. 18th ACM Conference on Information and Knowledge Management*, Hong Kong, November 2009.

[13] http://labs-repos.iit.demokritos.gr/skel/i-config/downloads/; accessed July 2010.

[14] Colbaugh, R. and K. Glass, "Predictive analysis for dynamical processes I: Multi-scale hybrid system modeling, and II: Predictability and warning analysis", *Proc. 2009 IEEE Intern. Multi-Conference on Systems and Control*, Saint Petersburg, Russia, July 2009.

[15] Lowd, D. and C. Meeks, "Good word attacks on statistical Spam filters", *Proc. Second Conference on Email and Anti-Spam*, Palo Alto, CA, July 2005.

[16] Cao, L., P. Yu, C. Zhang, H. Zhang, F. Tsai, and K. Chan, "Blog data mining for cyber security threats", *Data Mining for Business Applications*, Springer US, 2009.

[17] Colbaugh, R. and K. Glass, "Emerging topic detection for business intelligence via predictive analysis of 'meme' dynamics", *Proc. AAAI 2011 Spring Symposium*, Palo Alto, CA, March 2011.

[18] http://www.nongnu.org/ifile/; accessed July 2010.

# DISTRIBUTION

| | | | |
|---|---|---|---|
| 1 | MS0899 | Technical Library | 9536 (electronic copy) |
| 1 | MS0359 | D. Chavez, LDRD Office | 1911 |