2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation

Towards a Smart(er) Social Science using high-dimensional continuous-time trajectories from the Open Dynamic Interaction Networks (ODIN) platform

Bilal Khan
Department of Sociology
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
bkhan2@unl.edu

Kirk Dombrowski
Department of Sociology
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
kdombrowski2@unl.edu

Alekhya Bellam
Social Network Research Group
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
alekhyab505@gmail.com

Gisela Font Sayeras
Social Network Research Group
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
gfontsayeras@gmail.com

Kin Pi
Social Network Research Group
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
kinlampi@gmail.com

Devan Crawford
REACH Lab
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
dcrawford3@unl.edu

Patrick Habecker
REACH Lab
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
phabecker2@unl.edu

Maisha Jauernig
Social Network Research Group
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
mjauernig2@unl.edu

Abstract—In this paper, we describe Open Dynamic Interaction Networks (ODIN), a software platform designed to move the social, behavioral, and public health sciences toward a new investigative paradigm. ODIN enables us to collect and analyze rich, contextual continuous-time data on both personal change and interpersonal interaction. It achieves this by supporting dynamic delivery of questions based on the currently sensed context of each participant. The ODIN system extends beyond static (or even stepwise dynamic) graph-theoretic renderings of social life and individual behavior by considering "social relationship" to be measured in terms of high-dimensional continuous-time trajectories. The system is designed to be extensible, allowing seamless incorporation of new sensors, and correspondingly sophisticated compound rules by which contexts of interest may be specified. As such, ODIN opens the door for a "smarter" social science based on continuous contextual data, and a "smarter" data science that is reflective and sociologically informed.

Keywords—social network analysis, dynamic networks, longitudinal networks, temporal networks contextual survey, adaptive survey, ecological momentary assessment

I. Introduction

Invoking the distinction between an involuntary blink, a conspiratorial wink, and a satirical grimace, anthropologist Clifford Geertz reminded us that what people think they are doing ought to matter a lot in our understanding of what they are doing. Geertz's notion of "thick description" contrasts the recent rash of "thin" scientific analysis. Thin analysis is content to apply "machine learning" to "big data", presuming that the world operates by laws of "social physics". Criticism of these assumptions is common, and yet falls largely on deaf ears. Maintaining a commitment to the "meaning" of social-scientific data is increasingly difficult in large part because, to date, we lack the means for wide-scale, rapid, contextual and reflective data collection.

Traditional social network analysis (SNA) views human interactions as occurring over largely stable networks or sets of social connections.[1,2] This perspective allows SNA theorists to draw on graph theory as a basis for highly sophisticated social network analysis strategies.[3–5] Case in point is the development of Exponential Random Graph Modeling (ERGM),[6–9] which views social networks as formal logit-like statistical models whose weighted components are understood as the "logic" underlying link structure.[10]

The treatment of social network modeling in SIENA[11] follows a complementary approach. Here, models of *network change* are estimated using observations of the extant ties in a network at two different time points. With such data in hand, network change is simulated as a Markov process that influences individual actor states ("attributes") as well as the ties between actors.[6,12] Modelers select from a range of network-related mechanisms that could account for change within the network. They fit the mechanisms by simulating the network forward from time 1 to find parameter values that yield networks similar to those observed at time 2. The advantage of this approach is that it simultaneously accounts for both changes in network ties and actor states, and distinguishes attribute-changing "peer influence" from link-changing "homophily" dynamics.[7,13]

The ERGM and SIENA modeling approaches advanced our understanding of the process of individual and social change. In fact, the *network metaphor* for social interaction is largely invisible even as it grows immensely popular. However, at their root both approaches assume that the "observed" network is essentially stable, and the linking relationships are glossed as "equivalent" and "discrete".[8,9] Hidden in this process is that "network ties" in ERGM, SIENA—and SNA more generally—implicitly stand for ongoing relationships, which are themselves a selective abstraction of the more fluid world of interpersonal interaction. While the sedimentation of long term interactions into links allows network researchers to work with human relationships in a form that ordinary people find meaningful and substantive, a full picture of the richness and temporality

978-1-7281-4034-6/19/$31.00 ©2019 IEEE
DOI 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00051

of interactions is missing. As such, network renderings only capture “flattened”, freeze-frame pictures that compress complex relational and interactional dynamics into the identification of a fixed number of stable network alters.[2,14,15] In the majority of cases, this compression process goes unexamined and untheorized.

In contrast to these legacy paradigms, we carry out social inference directly on continuous-time data on the ever changing behaviors, attitudes, physiology, interactions, and experiences of a sample population over potentially long timescales. The Open Dynamic Interaction Networks (ODIN) platform is capable of providing such data in a reliable and scalable manner. The subject of this paper is design and validation of ODIN.

II. Prior Work

At present, no system/platform exists which provides the functionalities of the ODIN platform: (1) continuous-time social scientific data collection using cellphones, (2) web-based administration of a dynamic contextual survey instrument, (3) a specialized suite of analysis tools to work with continuous time social network data.

Software such as MobiClique (and before that BlueAware), and full platforms such as E-Small Talker use Bluetooth sensing data for purposes of social networking[16–18] (see Jabeur et al[19] for review). Many proposals for using Bluetooth sensor data to detect “social circles” recently appeared in computer-science related outlets.[20–23] However, these approaches use post-hoc analysis of interaction data, in the absence of any user-feedback or interpretation.

By far the most similar environment to ODIN is the *funf* open sensing framework, recently released by the MIT Media Lab (http://www.funf.org). While *funf* provides the integration of location, movement, communication and proximity data collection, it suffers from the same issues discussed above: it provides no means to elicit either static or adaptive/responsive data from the participant. This means any questions or responses associated with the *funf* data must be collected post hoc and from recall. The result is “data in a vacuum” that defaults to a social physics-like “thin description” that falls short of genuine explanatory power.

These problems go beyond theory: by limiting research to passively collected data, we cannot address important social and behavioral phenomena. For example, “thin” approaches of non-social/behavioral sciences model the transmission of social, behavioral, and economic phenomena as a one-way, passively received flow. The message sender can effect change in the receiver, but there is no dialogical exchange. Even when we account for varying probability of success (see Goyal[24]), physics-style diffusion models cannot account for polarization, which exists in class and community roles.[25] Simplified views of social interactions draw on and yield caricature conclusions, e.g. individuals with high degree bear the greatest power to influence.[26] As Dandekar[27] shows, to go beyond smooth and continuous models of social interaction, researchers must consider nonlinear models (e.g. processes involving “feedback” or “dialogue”) that account for influence and allow for clustering.

NSF project SMA1338485 saw the design of a responsive, contextual survey platform, involving collaborators from social, behavioral, and public health disciplines. Key design objectives included (i) an extensible on-phone system for engaging hardware sensors, (ii) a rich language for specifying the rules of participant prompts, (iii) analytic tools for making sense of the collected data. The ODIN system achieves these objectives; the implementation was funded by National Institutes of Health project R01GM118427.

The three components of the ODIN system are:

A. Mobile Platform: Each participant carries a cellphone on which a custom App continuously monitors the participant’s interactions, context, and state, and asks suitable questions.

B. Cloud-based Server: A dedicated machine in the cloud runs the web-based design of dynamic contextual social and behavioral health research protocols, and the secure aggregation of "live" data collected from participants.

C. Analytics Library: An extensible Python library helps analyze the continuous-time interaction and dynamic survey data collected at study participants' mobile platforms and aggregated by the ODIN Server.

Below we describe each of the three components.

III. ODIN’s Mobile Platform

Researchers recruit, screen and determine eligibility of participants. When a participant is enrolled in the study, they receive a unique 10-character alphanumeric coupon number. The participant downloads and installs the ODIN app onto their cellphone and launches the app. The app requires their assigned coupon number, and then contacts the ODIN server to download the study-relevant data, including: (i) information and consent forms, (ii) a list of sensors that the study engages, (iii) the fidelity and frequency with which each sensor will record data, (iv) study protocol questions, and (v) rules specifying the contextual circumstance(s) in which each question is asked. The ODIN app provides the participants with information about the study, and obtains their signature on study consent forms. Once the app has uploaded this signature to the ODIN server, the participant's app is considered active.

An active app engages two mandatory services: The Upload Service which periodically pushes sensor and answer data collected by the app up to the ODIN server; the RuleQuestion Service which periodically pulls updates to questions and rules from the ODIN server down to the app and determines whether the participant should be prompted with a question. Additionally, up to 4 (optional) services may be running, as depicted in **Figure 1**.

- The Location Service connects to the phone's GPS sensor, and periodically records the participant's latitude and longitude to the app’s Location database.
- The Proximity Service connects to the phone's Bluetooth low energy (BLE) sensor, and periodically records to the

National Institutes of Health, General Medical Sciences R01 GM118427.

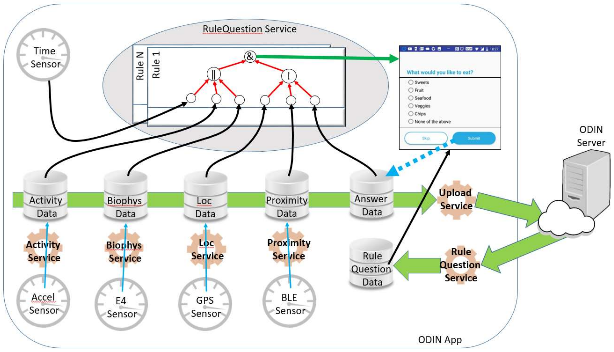

Figure 1. The architecture of the ODIN App

app's Proximity database the IDs and signal strength of all visible Bluetooth devices also running the ODIN app.

The Activity Service connects to the phone's Accelerometer sensor, and records its interpretation by using Google's Activity Recognition library, storing the conclusions to the app's Activity database.

- The Biophysiology Service connects to the Empatica E4 sensor via Bluetooth. The E4 provides a photoplethysmography sensor (to measure blood volume pulse at 64Hz and inter-beat interval, an electrodermal activity sensor operating at 4 Hz, and an optical thermometer to measure skin temperature at 4Hz.

Each of the four optional services interfaces with a specific sensor hardware. As part of defining an ODIN-enabled study, researchers must (i) specify the sensors/services they wish to enable within their study, and (ii) specify the frequency and of measurements for each sensor they choose to engage. **Table 1** shows the configuration parameters of each sensor.

Table 1. Service/Sensor Configuration

Service name Sensor name	Frequency Parameter
Location service **GPS sensor**	Sampling interval (seconds)
Proximity service **BLE sensor**	Sampling interval (seconds)
Activity service **Accelerometer sensor**	Sampling interval (seconds)
Biophysiology service **E4 sensor**	Sampling interval (seconds)

The ODIN app is architected so that researchers can readily add new "plugin" modules (consisting of a sensor, a service, and a data table). For example, we are currently integrating a Neuro service which interfaces to the Emotiv Insight wireless EEG headset.

A. The RuleQuestionService

The RuleQuestion Service periodically pulls updates to questions and rules from the ODIN server to the app. Most importantly, it determines whether the participant should be prompted with a question by monitoring the study-specific prompting rules. Each rule contains (i) a testable primitive based on the participant's current interactions, responses, and context, and (ii) a testable compound filter a based on the participant's *historic* interactions and responses. When a primitive is true, the associated compound filter is checked.If the filter is also true, the rule "fires", causing the app to display the associated question to the participant. **Table 2** lists the supported rule primitives.

Table 2. ODIN Rule Primitives

Names	Parameters
AtTime	CronSpec
OnArrival	Lat, Lon, Rad, minT, minF
OnDeparture	Lat, Lon, Rad, minT, minF
WhileAt	Lat, Lon, Rad, minTSLF
OnInteractionEnd	NumPeers
OnPhysiologyChange	minT, minF
OnActivityChange	ActName, minT, minF
OnButtonPress	ButtonPress

- AtTime (params: CronSpec) is true if the time now matches that given in the cron string CronSpec.
- OnArrival (params: Lat, Lon, Rad, minT, minF) is true if the most recent GPS sensor measurements show the participant is within Rad meters of location (Lat, Lon) for at least minT seconds, and was further than Rad meters from the location for at least minF seconds prior.
- OnDeparture (params: Lat, Lon, Rad, minT, minF) is true if the most recent GPS sensor measurements show that the participant was further than Rad meters of location (Lat, Lon) for at least minT seconds, and was closer than Rad meters from this location for at least minF seconds immediately prior.
- WhileAt (params: Lat, Lon, Radius, minTSLF) is true if the most recent GPS sensor measurement show the participant is closer than Rad meters from (Lat, Lon) and that the rule containing this primitive has not fired in the minTSLF ("min time since last fire") seconds prior.
- OnInteractionEnd (params: minT, minF) is true if the most recent BLE sensor measurements show that the participant was in the vicinity of no study participants for at least minF seconds, and was near at least 1 other study participant for at least minT seconds immediately prior.
- OnInactivity is true if the most recent Accelerometer sensor measurements show that the phone has been stationary for minT seconds, but was not stationary for at least minF seconds previously.
- OnPhysiologyChange (params: Sig, Perc) is true if the most recent E4 measurements show the participant's biophysiological Sig data has changed by Perc percent.
- OnButtonPress (params: BtnName) is true if the participant pressed the BtnName pushbutton inside their ODIN app.

A compound filter is a statement based on one or more elementary filters, joined using parentheses and logical connectives and (&&) or (||) and not (!). **Table 3** shows the supported elementary filters.

Table 3. Elementary Filters for Rules

Names	Parameters
isDateInRange	Dmin, Dmax
isTimeInRange	Tmin, Tmax
isInArea	Lat, Lon, Radius
IsInteracting	NumPeers

- isDateInRangeRange (params: Dmin, Dmax) is true if the date is between Dmin and Dmax.
- isTimeInRange (params: Dmin, Dmax) is true if the time now is between Dmin and Dmax.
- isInArea (params: Lat, Lon, Radius) is true if the last GPS sensor measurement shows the participant within Rad meters of location (Lat, Lon).
- isInteracting (params: NumPeers) is true if the last BLE sensor measurement shows the participant interacting with at least NumPeers many other participants.

IV. ODIN'S CLOUD-BASED SERVER

Four services operate within the ODIN server (see **Figure 2**)

- The Database Management System (DBMS) service maintains all data about studies, including study protocols and data collected from participants within each study.
- The Researcher REST service communicates with the DBMS, providing a REST API to a suite of abstract functions (e.g. create study, add a question to a study, make coupons for a study, get all answers collected from a participant, etc.)
- The Researcher webservice communicates with the Researcher REST service, and serves dynamic Javascript and HTML pages following an MVC design.
- The PhoneApp service mediates between the ODIN app instances running on participant phones, and the DBMS on the server, providing a REST API to a suite of abstract functions (e.g. send signed consent, send participant answers, request updates to questions and rules, etc.)

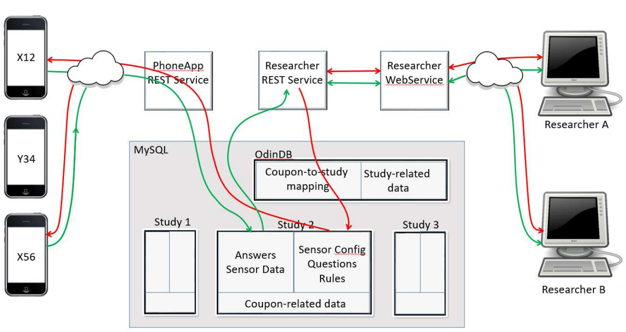

Figure 2. The architecture of the ODIN Server

V. RESEARCHER WORKFLOW

An ODIN-enabled study passes through three distinct phases, each of which requires/supports distinct actions on the part of the researcher; these phases are described below. While **preparing** a study, the researcher may:

- Specify sensors to be engaged, as well as fidelity and frequency of sensor recordings;
- Specify study information and consent forms to be provided to participants;
- Specify questions that make up the study protocol;
- Specify contextual rules that govern when each question will be asked;

While the Study is **in progress**, the researcher may:

- Generate unique coupons to be distributed;
- Distribute coupons to eligible recruits, who download the ODIN app, and enter the coupon number, and provide informed consent;
- Examine data being collected from study participants on an ongoing basis;
- Create new questions and rules;
- Disable existing questions and rules;
- Drop participants from the study on the basis of poor responsiveness.

Current participants may decide to withdraw consent using the app, in which case no further data is collected from them. Once the study is **completed**, the ODIN system stops collecting data, and the researcher may:

- No longer enroll further participants;
- No longer modify questions and rules;
- Examine and analyze data that was collected from study participants. Supported analyses are described in the next section.

VI. ODIN'S ANALYTICS LIBRARY

Analytic support is provided through an extensible library of Python routines that provide researchers with tools to integrate and interpret data collected by ODIN. The data processing pipeline involves a sequence of two step.

In Step 1, the researcher defines a set of K coordinate functions $F_1, F_2, \ldots F_K$, that can be evaluated for any study participant at any point in time; these functions are of one of six types:

1. Location-based functions use data from the GPS sensor e.g. F_1: "Is this participant within 200 meters of their home right now? (1=yes, 0=no)"; F_2: "The number of times this participant was within 200 meters of a specific bar in the past 7 days";
2. Proximity-based functions use data from the BLE sensor, e.g. F_3: "Is this participant near any other study participants right now, and if so, how many?"; F_4: "The average duration of interactions that this participant engaged in with other study participants, over the past 14 days".

3. Activity-based functions use data from the Accelerometer sensor, e.g. F_5: "Is this participant active right now? (1=yes, 0=no)"; F_6: "The fraction of time the participant has been inactive over the last 4 hours".
4. Biophysiology-based functions use data from the E4 sensor, e.g. F_7: "The heart rate of this participant right now"; F_8: "The average heart rate of this participant over the last 24 hours".
5. Answer-based functions use data from previously answered questions, e.g. F_9: "Did this participant answer *yes* to question #2 concerning whether they felt any craving for alcohol? (1=yes, 0=no)"; F_{10}: "The average number of times the participant answered *yes* to multiple choice question #2, over the entire duration of their participation in the study so far".
6. Compound functions are built from the previous five primitive types, e.g. F_{11}: "The average duration for which this participant interacted with another participant whose most recent answer to question #2 differed, while both were within 200 meters of a specific bar".

Because coordinate functions of types 1-4 reference a sensor, and each sensor only provides data at a researcher-specified sampling frequency, some functions may not be defined at all points in time. For example, function F_1 is only defined at times when the GPS sensor measurement is made; F_3 is only defined at times when Bluetooth-based proximity is recorded; F_5 is only defined at times when the accelerometer is being measured; F_7 is only defined at times when the E4 sensor is being polled. Similarly, F_9 may not be defined if the participant was never asked question #2. Coordinate functions are called discrete-time if they are not defined at all times. Likewise, because functions F_2, F_4, F_6, F_8, and F_{10} aggregate measurements over intervals of time, they too may be undefined if occasionally no measurements occur within those time windows.

In Step 2, we convert discrete-time coordinate functions to continuous-time functions that are defined at all times. To facilitate this, the researcher specifies a data imputation strategy for each discrete-time coordinate function. This allows the Analytics Platform to impute missing values, and ensure that all coordinate functions are defined at all points in time. Four data imputation strategies are supported:

- Categorical-Nearest-Value: The missing value is filled in using *either* the previous known value *or* the subsequent known value, *whichever is most temporally proximate*. We illustrate this strategy applied to F_1: Suppose there was a GPS sensor reading at 9:00 and 10:00, and the participant was seen to be at home early on (i.e. F_1=1 at 9:00), and far from home later (i.e. F_1=0 at 10:00). Applying this completion strategy would fill in the missing values from 9:01 to 9:30 with F_1=1, and from 9:31 to 9:59 with F_1=0. The implicit assumption being made is that the categorical value of the function changes at the halfway point between successive readings of the sensor.
- Copy-Last: The missing value is filled in using the most temporally proximate *previous* known value. We illustrate this strategy applied to F_3: Suppose there was a BLE sensor reading at 11:00 and 11:15, and the participant was seen to be not interacting with anyone early on (i.e. F_3=0 at 11:00), and then interacting with 2 other participants later (i.e. F_3=2 at 11:15). Applying this completion strategy would fill in the missing values from 11:00 to 11:14 with F_3=0. The implicit assumption being made is that the value of the function changes at exactly the time that the new reading is made from the sensor.
- Linear-Interpolate: The missing value is filled in using proportional average of the most temporally proximate previous and subsequent known values. We illustrate this strategy applied to F_7: Suppose there was an E4 sensor reading at 14:15 and 14:20. Suppose the participant had a heart rate of 64 bpm earlier (i.e. F_7=64 at 14:15) and 74 bpm later (i.e. F_7=74 at 14:20). Applying this completion strategy would fill in the missing values as follows: F_7=66 at 14:16; F_7=68 at 14:17; F_7=70 at 14:18; F_7=72 at 14:19. The implicit assumption being made is that the function changes smoothly over the interval between successive measurements of the sensor.
- Copy-Last-With-Decay (D_{val}, T_{max}): The missing value is filled in using the most temporally proximate previous measurement, unless that measurement is longer than T_{max} in which case the default value D_{val} is used. We illustrate this strategy applied to F_9, with $D_{val}=0$ *and* T_{max} = 1 hour. Suppose question #2 was answered positively at 17:05 pm and then answered negatively at 23:10 pm (i.e. F_9=1 at 17:05 and F_9=0 at 23:10). Applying this completion strategy would fill in the missing values from 17:06 to 18:05 with F_9=1, and from 18:06 to 23:09 with F_9=0. The implicit assumption being made is that the value of the function reverts to a default value some fixed interval after a sensor reading is made.

Once the researcher has completed the two step process, the Analytics Platform map each participant's ODIN data to high-dimensional continuous-time trajectory in K-dimensional space. This high-dimensional continuous-time data is then able to be subjected to a number of analyses, described below.

A. Trajectory Dynamics Modeling

To understand the dynamics of participant experience over the course of a study, Vector Autoregression (VAR) modeling is engaged. VAR models have been used successfully in many healthcare settings. For example, by modeling the co-evolution of heart rate, respiratory rate, and pulse oximetry, Bose et al. found that there was a relationship between the signals changed during physiologic stress and the change forecast impending cardiorespiratory instability[28]. In another example, the dynamic (bidirectional) causal relationships between physical activity and affect was quantified using VAR by Stavrakakis et al, who found significant clustering in inter-individual variation[29]. Following these examples, we use VAR because it has demonstrates good forecasting capability in bio-social data applications, and supports testing for Granger causality[30]. Approach: In our application of VAR, each of the K variables is successively considered to be focal; the evolution (imminent change) of the focal variable is then modeled as a linear function of: (i) its own lagged values, and (ii) lagged values of up to L of the other $K\text{-}1$ variables. Each regression involves $L+1$ predictors variables at 1 lagged time. In practice, we took L=3 to limit model complexity. For each choice of focal variable, we exhaustively tried all size-*3* subsets of the K variables as potential predictors. For each size-*3* subset considered, we found optimal (maximally predictive) lag times using the AutoVAR[31] software, which automates this search process[32] by evaluating all possible VAR models within the combinatorial search space. Power: A standard power analysis of multilinear regression with $L+1$=4 predictors shows that statistically significant determination of regression coefficients is achieved at effect sizes of 0.15 (taking significance/type I error α=0.05, type II error rate β=0.10, power=0.90), provided at least 84 data points are available. In ODIN-enabled studies where there is an extended participation period (i.e. over 3 months), and measurement frequency is moderate (i.e. once a day or more), this amount of data (84 data points) would be generated by *each participant*, allowing for the micro-timescale modeling of the trajectories of each individual separately. In studies where duration of participation is shorter (i.e. less than 3 months), or measurement frequency is low (i.e. less than once a day), the data from multiple participants must be aggregated to uncover trajectory dynamics at the level of entire sub-populations. Modeling is carried out using the *svar* package in SciPy[33]. Outputs: For each participant (or sub-population), we output a VAR model quantifying the co-evolution dynamics of each the K coordinates as linear functions of lagged values of their respective best 3 predictors. Validation: These K regressions are evaluated using standard measures of validity such as R^2 and analysis of residuals.

B. *Learning Trajectory Classifiers*

This second type of analysis can be engaged once Trajectory Dynamics Modeling is complete (see previous section). To start, the researcher provides a classification of participant trajectories using discrete labels. For example, they might label all trajectories of females as "Type F", and those of male participants as "Type M". Alternately, they might label all trajectories of participants of individuals who appeared to experience an alcohol abuse relapse as "Type 1", and those who did not as "Type 2". The process by which each participant's continuous-time trajectory is labelled is beyond the scope of the Analytics Platform. In our approach, the VAR model coefficients from the Trajectory Dynamics Modeling process (described in the previous section) are taken as predictor variables for each individual, while the researcher provided label is the dependent variable to be learned. At present, we only consider decision-tree classifiers because of their direct interpretability by researchers. The decision tree via the C4.5 algorithm using SciPy[33].

C. *Unsupervised Trajectory Clustering*

A final type of analysis involves mining continuous trajectory data to find clusters of similarity trajectories. This "unsupervised" analysis requires searching for subspaces of lower dimension ($\leq K$) in which participants' trajectories project into well-separated clusters. Clusterings imply the discovery of new typologies of participant experience based on quantitative longitudinal characteristics. To achieve this, we draw on recent advances in trajectory mapping.[34,35] At its core, trajectory clustering requires a measure of the similarity between two trajectories that allows for varying amounts of time warping (shifting and dilation) to align corresponding trajectory points.[36–38] Given a similarity measure, we bundle proximate trajectories into "bundles" using agglomerative clustering[39]. To search efficiently for low dimensional ($\leq K$) subspaces in which the (projected) trajectories are found to fall into a few, well-separated clusters, we apply the density-based approach of the INSCY algorithm[40], finding clusters recursively as dense regions separated by sparse areas.

VII. System Validation Results

To evaluate the reliability of sensor data, we conducted an experiment used a benchmark study in which all 4 sensors were enabled. Each sensor was set to record at an interval of 5 minutes. Two phones of different brands were registered to the study: A Motorola Moto G, and a Huawei Nexus 6P. The phones participated in the study for 72 hours, during which they remained stationary and at a fixed location, 4 meters apart. At the end of this period, we evaluated the data from sensors in 3 distinct ways: (i) What percentage of the sensor readings were missing? (ii) What was the mean error in the temporal spacing of successive readings relative to the expected 5 minutes? (iii) What was the mean error in the values of the sensor readings? The results of this experiment are shown in **Table 3**. The biophysiology sensor's mean value error is not reported because we lacked a reference signal, and the hardware vendor did not provide error rate information.

Table 3. Sensor Reliability

Sensor	Percent Values Missing	Mean Temporal Error	Mean Value Error
Location	<5%	~30 sec	10 meters
Proximity	<5%	~30 sec	±6 dB
Activity	<5%	~30 sec	None
Biophysiology	<5%	~30 sec	Unknown

To evaluate the reliability of AtTime rules we conducted a 72-hour experiment involving aforementioned 3 phones, in which a question was to be asked every hour on the hour. False negatives were noted 4% of the time, if the rule failed to fire within 5 minutes of the time expected. False positives were observed 7% of the time, if the rule fired when no firing was expected within ±5 min.

To evaluate the reliability of OnArrival, OnDeparture, and WhileAt we conducted a 72-hour experiment involving aforementioned 3 phones, placing all 3 within a 200m radius region of interest from 9-10am, 11-noon, 1-2pm, 3-4pm, and then moved them out of the region between 10-11am, noon-1pm, 2-3pm, 4-5pm. False negatives were noted 6% of the time, if the rule failed to fire within 5 minutes of the time expected. False positives were observed 8% of the time, if the rule fired when no firing was expected within ±5 min.

To evaluate the reliability of OnInteractionEnd rules, we conducted a 72-hour experiment involving 3 phones (Motorola Moto G, Huawei Nexus 6P, Motorola Moto E). Interaction was gauged based on physical proximity between participants, using a cutoff of 3 meters. Phones were placed in a collinear configuration, with phones #1 and #2 at a separation of ~2 meters, and phones #2 and #3 at 5-meter separation. Proximity was estimated using RSSI signal strength using a cutoff of 70 dB corresponding to 3 meters. False negatives were observed 12% of the time, when two phones close together but failed to report the interaction. False positives were observed 11% of the time, when two phones were not close together reported an interaction.

To evaluate the reliability of OnInactivity rules, we conducted a 6-hour experiment involving aforementioned 4 phones, of which 2 were stationary on a table, and 2 were carried by a researcher. False negatives were observed 10% of the time, when a phone being carried reported that it was stationary. False positives were observed 0% of the time, when a phones that was stationary reported being active.

The false positive and false negative rates for each of the 8 types of rules are given in **Table 4**. All false positives and negatives were found to be attributable to the fact that the Android operating system does not support real-time guarantees, and phone vendors often institute aggressive hardware sleep policies to preserve battery life. GPS, Bluetooth, and accelerometer sensor hardware failures did not contribute significantly to rule error rates. It was not possible to evaluate the OnPhysiologyChange rules because we lacked the ability to generate reference signals. OnButtonPress did not experience any false positive or negative events.

Table 4. Rule Reliability

Rule Primitive	False Negative Rate	False Positive Rate
AtTime	4%	7%
OnArrival OnDeparture WhileAt	6%	8%
OnInteractionEnd	12%	11%
OnInactivity	10%	0%
OnPhysiologyChange	Unknown	Unknown
OnButtonPress	0%	0%

VIII. A Small Pilot Study

A prototype ODIN implementation was validated over a 3-week pilot study that sought to simultaneously acquire continuous time trajectory data from 19 students in one section of an undergraduate mathematics course. During this time, contextual data on subjects' aptitudes and interactions were collected using the ODIN platform. **Figure 3** shows a graphical rendering of a one-dimensional projection of the trajectories (one trajectory for each of the 19 students). The x-axis represents time across an 8-hour business day, while the y-axis represents the number of interactions that each participant had in the past 30 minutes (with other study participants). One observes that in the initial 90 minutes almost all trajectories have high values (~18) corresponding to the 9am course lecture schedule. The fluctuation in values over the lunchtime and afternoon hours reflects both student sociality and shared class schedules.

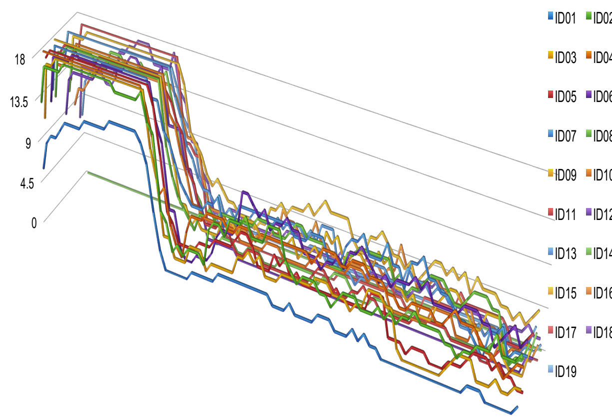

Figure 3. Trajectory Data from a Pilot Study

IX. Conclusions and Future Work

Social relationships and individual behaviors are known to co-evolve. The ability to distinguish between homophily (the formation of social links due to matching individual traits) and diffusion is a critical part of any research on influence[44] and behavioral epidemiology. Unfortunately, fine-grained temporal data on behaviors and interactions is rarely available[41]. Prior to ODIN (http://odin-software.com), investigations seeking to discern homophily and diffusion processes were far more laborious (e.g. Urberg's study[42] and Bauman et al.'s study[43] on smoking, where longitudinal design enabled separation of the effects of peer influence from those of selective association).

Geertz's challenge to engage in "thick description" of social phenomena requires investigators to move beyond purely observational data collection. The scale of data does not compensate for the inadequacy of purely passive modes of collection. The ODIN platform provides not only fine-grained temporal, spatial, social, and physiological data, but also has the capacity to query how participants understand their social interactions in that moment. The responsive and adaptive question designs paired with the passive data collection capacities of ODIN bring us closer to achieving a "thick description" of social phenomena, and reduce our reliance on "thin" scientific analysis.

Our future work will focus on supporting a wider array of sensors, rules, and improving the reliability of sensor operation and the rule evaluation system. ODIN's design and development was funded by National Institutes of Health grant R01GM118427.

REFERENCES

[1] Butts, C. T. Revisiting the foundations of network analysis. Science 325, 414 (2009).

[2] Bernard, H. R., Killworth, P., Kronenfeld, D. & Sailer, L. The Problem of Informant Accuracy: The Validity of Retrospective Data. Annual Review of Anthropology 13, 495–517 (1984).

[3] Carrington, P. J., Scott, J. & Wasserman, S. Models and methods in social network analysis. (Cambridge university press, 2005).

[4] Wasserman, S. Social network analysis: Methods and applications. (Cambridge university press, 1994).

[5] Wasserman, S. & Robins, G. An introduction to random graphs, dependence graphs, and p*. Models and methods in social network analysis 148–161 (2005).

[6] Snijders, T. A. B., van de Bunt, G. G. & Steglich, C. E. G. Introduction to stochastic actor-based models for network dynamics☆. Social Networks 32, 44–60 (2010).

[7] Veenstra, R., Dijkstra, J. K., Steglich, C. & Van Zalk, M. H. W. Network–Behavior Dynamics. J Res Adolesc 23, 399–412 (2013).

[8] Abraham, A. & Hassanien, A. E. Computational social network analysis: Trends, tools and research advances. (Springer, 2010).

[9] Kolaczyk, E. D. Statistical Analysis of Network Data: Methods and Models. (Springer, 2010).

[10] Wasserman, S. & Pattison, P. Logit models and logistic regressions for social networks: I. An introduction to Markov graphs andp. Psychometrika 61, 401–425 (1996).

[11] Ripley, R. M., Snijders, T. A. & Preciado, P. Manual for RSIENA. University of Oxford: Department of Statistics, Nuffield College (2011).

[12] Steglich, C., Snijders, T. A. . & West, P. Applying SIENA: An Illustrative Analysis of the Coevolution of Adolescents' Friendship Networks, Taste in Music, and Alcohol Consumption. Methodology: European Journal of Research Methods for the Behavioral and Social Sciences 2, 48 (2006).

[13] Steglich, C., Snijders, T. A. & Pearson, M. Dynamic networks and behavior: Separating selection from influence. Sociological methodology 40, 329–393 (2010).

[14] Butts, C. T. Social network analysis: A methodological introduction. Asian Journal of Social Psychology 11, 13–41 (2008).

[15] Killworth, P. D. & Bernard, H. R. Informant accuracy in social network data III: A comparison of triadic structure in behavioral and cognitive data. Social Networks 2, 19–46 (1979).

[16] Pietiläinen, A.-K., Oliver, E., LeBrun, J., Varghese, G. & Diot, C. MobiClique: middleware for mobile social networking. in Proceedings of the 2nd ACM workshop on Online social networks 49–54 (ACM, 2009).

[17] Zhang, R., Zhang, Y., Sun, J. & Yan, G. Fine-grained private matching for proximity-based mobile social networking. in INFOCOM, 2012 Proceedings IEEE 1969–1977 (IEEE, 2012).

[18] Champion, A. C. et al. E-SmallTalker: A distributed mobile system for social networking in physical proximity. Parallel and Distributed Systems, IEEE Transactions on 24, 1535–1545 (2013).

[19] Jabeur, N., Zeadally, S. & Sayed, B. Mobile social networking applications. Communications of the ACM 56, 71–79 (2013).

[20] Zheng, J. & Ni, L. M. An Unsupervised Learning Approach to Social Circles Detection in Ego Bluetooth Proximity Network. in Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing 721–724 (ACM, 2013). doi:10.1145/2493432.2493512

[21] Eagle, N. & (Sandy) Pentland, A. Reality Mining: Sensing Complex Social Systems. Personal Ubiquitous Comput. 10, 255–268 (2006).

[22] Stopczynski, A. et al. Measuring Large-Scale Social Networks with High Resolution. PLoS ONE 9, e95978 (2014).

[23] Eagle, N., Pentland, A. (Sandy) & Lazer, D. Inferring friendship network structure by using mobile phone data. PNAS 106, 15274–15278 (2009).

[24] Goyal, A., Bonchi, F. & Lakshmanan, L. V. S. Learning Influence Probabilities in Social Networks. in Proceedings of the Third ACM International Conference on Web Search and Data Mining 241–250 (ACM, 2010). doi:10.1145/1718487.1718518

[25] Chou, B.-H. & Suzuki, E. Discovering Community-Oriented Roles of Nodes in a Social Network. in Data Warehousing and Knowledge Discovery (eds. Pedersen, T. B., Mohania, M. K. & Tjoa, A. M.) 52–64 (Springer Berlin Heidelberg, 2010).

[26] Barabási, A.-L. The network takeover. Nature Physics 8, 14–16 (2012).

[27] Dandekar, P., Goel, A. & Lee, D. Biased Assimilation, Homophily and the Dynamics of Polarization. arXiv:1209.5998 (2012).

[28] Bose, E., Hravnak, M. & Sereika, S. M. Vector Autoregressive (VAR) Models and Granger Causality in Time Series Analysis in Nursing Research: Dynamic Changes Among Vital Signs Prior to Cardiorespiratory Instability Events as an Example. Nurs Res 66, 12–19 (2017).

[29] Stavrakakis, N. et al. Temporal dynamics of physical activity and affect in depressed and nondepressed individuals. Health Psychology 34, 1268–1277 (2015).

[30] Kirchgässner, G., Wolters, J. & Hassler, U. Granger Causality. in Introduction to Modern Time Series Analysis (eds. Kirchgässner, G., Wolters, J. & Hassler, U.) 95–125 (Springer Berlin Heidelberg, 2013). doi:10.1007/978-3-642-33436-8_3

[31] Emerencia, A. C. et al. Automating Vector Autoregression on Electronic Patient Diary Data. IEEE Journal of Biomedical and Health Informatics 20, 631–643 (2016).

[32] van der Krieke, L. et al. Ecological Momentary Assessments and Automated Time Series Analysis to Promote Tailored Health Care: A Proof-of-Principle Study. JMIR Res Protoc 4, (2015).

[33] Bressert, E. SciPy and NumPy. (O'Reilly Media, Inc., 2012).

[34] Li, Z. et al. MoveMine: mining moving object databases. in Proceedings of the 2010 ACM SIGMOD International Conference on Management of data 1203–1206 (ACM, 2010).

[35] Wu, F., Lei, T. K. H., Li, Z. & Han, J. MoveMine 2.0: Mining object relationships from movement data. Proceedings of the VLDB Endowment 7, (2014).

[36] Rabiner, L. R. & Juang, B.-H. Fundamentals of speech recognition. 14, (PTR Prentice Hall Englewood Cliffs, 1993).

[37] Vlachos, M., Kollios, G. & Gunopulos, D. Discovering similar multidimensional trajectories. in Data Engineering, 2002. Proceedings. 18th International Conference on 673–684 (IEEE, 2002).

[38] Piciarelli, C. & Foresti, G. L. On-line trajectory clustering for anomalous events detection. Pattern Recognition Letters 27, 1835–1842 (2006).

[39] Hastie, T. et al. The elements of statistical learning. 2, (Springer, 2009).

[40] Assent, I., Krieger, R., Muller, E. & Seidl, T. INSCY: Indexing Subspace Clusters with In-Process-Removal of Redundancy. in Eighth IEEE International Conference on Data Mining, 2008. ICDM '08 719–724 (2008). doi:10.1109/ICDM.2008.46

[41] Urberg, K. A., Luo, Q., Pilgrim, C. & Degirmencioglu, S. M. A two-stage model of peer influence in adolescent substance use: individual and relationship-specific differences in susceptibility to influence. Addictive Behaviors 28, 1243–1256 (2003).

[42] Urberg, K. A. Locus of peer influence: Social crowd and best friend. J Youth Adolescence 21, 439–450 (1992).

[43] Bauman, K. E. & Fisher, L. A. On the measurement of friend behavior in research on friend influence and selection: Findings from longitudinal studies of adolescent smoking and drinking. J Youth Adolescence 15, 345–353 (1986).

[44] Steglich, C., Snijders, T. A. & Pearson, M. Dynamic networks and behavior: Separating selection from influence. Sociological methodology 40, 329–393 (2010).

[45] Dong, W., Lepri, B. & Pentland, A. (Sandy). Modeling the Co-evolution of Behaviors and Social Relationships Using Mobile Phone Data. in Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia 134–143 (ACM, 2011). doi:10.1145/2107596.2107613

[46] Weng, L. et al. The Role of Information Diffusion in the Evolution of Social Networks. in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 356–364 (ACM, 2013). doi:10.1145/2487575.2487607