THE GEOGRAPHY OF ANONYMOUS COMMUNICATIONS: PREDICTING ESCALATION OF ANONYMITY NETWORKS DURING EVENTS OF CIVIL UNREST

by

Brian Sandberg
A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Master of Science
Geoinformatics and Geospatial Intelligence

Committee:

Dr. Anthony Stefanidis, Thesis Director
Dr. Arie Croitoru, Committee Member
Dr. Dieter Pfoser, Committee Member
Dr. Dieter Pfoser, Department Chairperson
Dr. Donna M. Fox, Associate Dean, Office of Student Affairs & Special Programs, College of Science
Dr. Peggy Agouris, Dean, College of Science

Date: Spring Semester 2018
George Mason University
Fairfax, VA
The Geography of Anonymous Communications: Predicting Escalation of Anonymity Networks During Events of Civil Unrest

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

by

Brian Sandberg
Bachelor of Science
Kent State University, 1990

Director: Anthony Stefanidis, Professor
Department of Geography and Geoinformation Science

Spring Semester 2018
George Mason University
Fairfax, VA
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>iv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>v</td>
</tr>
<tr>
<td>Abstract</td>
<td>vi</td>
</tr>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Literature Review</td>
<td>4</td>
</tr>
<tr>
<td>Data Preparation</td>
<td>15</td>
</tr>
<tr>
<td>Methodology</td>
<td>27</td>
</tr>
<tr>
<td>Predictive Modeling</td>
<td>27</td>
</tr>
<tr>
<td>Algorithm Selection and Hyperparameter Tuning</td>
<td>29</td>
</tr>
<tr>
<td>Automated Machine Learning</td>
<td>37</td>
</tr>
<tr>
<td>Results</td>
<td>40</td>
</tr>
<tr>
<td>Discussion</td>
<td>45</td>
</tr>
<tr>
<td>Conclusion</td>
<td>50</td>
</tr>
<tr>
<td>References</td>
<td>51</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1. Selected Countries.</td>
<td>16</td>
</tr>
<tr>
<td>Table 2. ACLED Event Types and Actor Categories.</td>
<td>17</td>
</tr>
<tr>
<td>Table 3. Summary of Classification Models and Model Hyperparameters</td>
<td>36</td>
</tr>
<tr>
<td>Table 4. Predictive Modeling Results.</td>
<td>41</td>
</tr>
<tr>
<td>Table 5. Automated Machine Learning Results.</td>
<td>42</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1. Time Series Plots for ACLED Events and Tor Usage.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 2. Sample Time Series Plots.</td>
<td>23</td>
</tr>
<tr>
<td>Figure 3. Sample Event-based Spatial Clusters.</td>
<td>25</td>
</tr>
<tr>
<td>Figure 4. Input Features for the Classification Tasks.</td>
<td>26</td>
</tr>
<tr>
<td>Figure 5. Predictive Modeling Workflow.</td>
<td>29</td>
</tr>
<tr>
<td>Figure 6. Automated Machine Learning Workflow.</td>
<td>38</td>
</tr>
</tbody>
</table>
ABSTRACT

THE GEOGRAPHY OF ANONYMOUS COMMUNICATIONS: PREDICTING ESCALATION OF ANONYMITY NETWORKS DURING EVENTS OF CIVIL UNREST

Brian Sandberg, M.S.
George Mason University, 2018
Thesis Director: Dr. Anthony Stefanidis

Civil unrest can trigger escalation of anonymous communication to conceal user identity for protection or to circumvent censorship. Anonymization tools such as the Tor Network can support planning, orchestrating, or responding to protest events, while hiding user’s real location and identity. This research aimed to understand the relationship between protest events and Tor usage. Protests may be spontaneous or planned, and the affordances of anonymity networks may involve one or two day time lags preceding or following an event occurrence. Accurately characterizing this relationship required accounting for these different temporal usage patterns.

A methodology was developed to automatically discover the best estimator for predicting Tor usage in response to protest events. Different classifiers were fit to data representing the different events types, number of events, actor categories, fatalities, and Tor metrics. It was also of interest to understand the use of anonymizing technology
against the backdrop of different government regimes. While protests in democracies may be conducted overtly, those in authoritarian regimes may require more covertness by participants.

Experiments were conducted using over five years of conflict event and Tor usage data. Nine countries were selected from the Armed Conflict Location and Event Dataset (ACLED), which focuses on countries in Africa. A selection criterion was based on large populations with Internet penetration rate of at least 25%, and an active history of protest events that occurred over the time period of the study. In addition, an even distribution of countries was selected based on their designation of free, partly-free, and not-free by the Freedom House organization. This research produced unique quantitative results demonstrating the contribution of anonymity networks to the formation and functioning of social movements and collective behavior. Prediction F1 Score was over 86%, which indicated a strong signal existed between civil unrest events and Tor usage. Results are significant given the multitude of use cases for Tor, with consistent escalation occurring during protest events, particularly in authoritarian regimes.
INTRODUCTION

The geography of cyberspace changes the traditional analytic landscape of the natural and built environment to a virtual environment. As with geographic analysis in physical space, analysis of cyberspace is equally important to understand patterns of activity and behavior that are emerging, evolving, or changing over time (Lee & Chan, 2016). In cyberspace, many geographic-based assumptions and estimates no longer hold true. Patterns and relationships can emerge independent of geographic location. For instance, the idea that near things are more related than distant things has been a core concept in geographic analysis (Tobler, 1970). Transportation systems, communication systems, and the Internet have altered this assumption and have created new challenges and opportunities for spatial analysis. Janelle (1969) described how innovations in transportation systems had a major impact on human movement and interaction. Cyberspace further removes the dependence on distance and geographic space for connections and collective behaviors to converge. Social cliques can include participants with similar interests from diverse geographies (Nicholls Walter, 2008). Social networks and cybercrime networks bring together participants from distant places (Leukfeldt, Kleemans, & Stol, 2017). Periods of civil unrest unite participants from near and far, and in these cases, the physical and virtual worlds intersect (Braha, 2012).
Political events, including protests and riots, are supported via social networks, mobile communications, and anonymization tools. It is often desirable to measure these virtual activities to understand patterns relating to real world events. Brand marketers are interested in online consumer trends and habits to understand audience engagement and advertising effectiveness. Governments and militaries need to understand how social networks are used by terrorist groups to facilitate propaganda, radicalization, and recruitment activities (Klausen, 2015). In this latter case, actors often obscure their identities to conduct online activities anonymously. There are also legitimate reasons to obscure online identity, including digital privacy and consumer protections, such as avoidance of scams and identity theft (Rika Butler, 2007). Military, law enforcement, investigative journalists, and political dissidents also have strong motivations to use anonymizing technology to protect their informants, sources, and their own identity. The exclusion of user identity and physical location can present some challenges when studying the geography of anonymous communication. While analyzing individual usage patterns of anonymity networks is generally not possible, it is feasible to analyze and draw conclusions on aggregate behaviors of anonymous users.

On a daily basis, the various news media report on different types of civil unrest occurring throughout the world. These collective action events range from peaceful protests to violent uprisings. Every day people are also using anonymization technology while communicating and sharing information online to hide their real identity, location, and activity. This study analyzes and measures patterns of collective behavior involving the use of anonymity networks to facilitate social movements. This research aimed to
determine if there is a relationship between protest events and the use of anonymization technology. In other words, what is the effect of civil unrest on the use of anonymizing tools? The hypothesis is that protests increase the use of anonymizing tools and the null hypothesis is that no significant change in usage patterns exist prior to or during protests. Is the relationship more pronounced in authoritarian regimes where public protests may be illegal or subject to military response? The goal was to produce quantitative analytic results that demonstrate that technologies like the Tor Network are used for important societal purposes such as opposing repressive governments or circumventing censorship, rather than just criminal activities such as terrorist recruiting, drug marketplaces, and child abuse material.
LITERATURE REVIEW

The literature review covers three areas of related research starting with the evolution of crisis event databases and the improved quality to enable the type of research conducted in this study. The review continues with the affordances of technology, such as mobile communications and social media, to civil unrest and protest event, and how computers, the Internet, and particularly, anonymity networks have provided support to protests and social movements. The literature review concludes with key research on automated machine learning and hyperparameter optimization, which are the underlying technologies needed to develop an automated protest event prediction capability using protest event data to predict escalation in Tor usage.

Development of systematic, geographical disaggregated event datasets have helped foster progress in conflict research (Gleditsch, Metternich, & Ruggeri, 2014). Crisis event data has evolved over the years and specialized databases now exist for social and political science research and for a variety of hypothesis testing. These datasets record a “who did what to whom” scheme within the context of political actions. This observational data is a record of individual events between a source actor and target actor and provides a disaggregated view of political events. Modern automated machine coded datasets, such as Integrated Crisis Early Warning System (ICEWS), Global Database of Events, Language, and Tone (GDELT), and Phoenix, are based on news sources and provide insights into what has been reported and collected (Arva et al., 2013). There are several idiosyncratic aspects of these datasets that complicates analysis. A trivial event
can attract huge media attention, while major political events may not be reported adequately. Reports may also be false, incomplete, contain duplicates, or have incorrect coding. The Armed Conflict Location and Event Dataset (ACLED) provides a more focused, geographically disaggregated dataset for analyzing protest events and provides a more accurate specification for the locations and dates of the observed events (Raleigh, Linke, Hegre, & Karlsen, 2010).

Most research using event data has focused on monitoring and mining content for emerging trends, and processing trends into forecasts of political events. Quantitative analysis of crisis event data has also focused on predicting event onset. Ramakrishnan et al. (2014) describes how Early Model Based Event Recognition using Surrogates (EMBERS) outperforms existing methods in event forecasting of civil unrest using open source indicators (OSI) from social media, news events, blogs, and economic indicators. Muthiah (2014) provides a more detailed investigation into this EBERS framework, while Agarwal and Sureka (2015) provides a review of related literature. Related studies have analyzed the geography of civil unrest. Buhaug and Gates (2002) analyze geographical factors that determine the scope of conflict and the location of the conflict relative to the capital. Sensitivity analysis of empirical results on conflict onset is covered in Hegre and Sambanis (2006), with findings that confirm large population and low per capita income increase risk of civil war. Schutte and Donnay (2014) present a causal analysis that reveals that Iraqi civilians actively supported US military in reaction to indiscriminate insurgent violence.
Protests are a specific class of conflict events and one in which cyberspace has played a significant role. Information sharing and communication technologies are critical to planning and organizing social movements. Different forms of communication, including social media and anonymous communications, can provide unique benefits in facilitating these events. Yuan (2017) discussed the use of mass media and location-based social media data and shows a strong distance decay effect on connections and interactions between Chinese provinces. Bastos, Recuero, and Zago (2014) investigated the relationship between onsite and online protesting activity, concluding that online participants are geographically distant from street protesters. Conover et al. (2013) examined how the goals of a protest movement are reflected in the geographic patterns and information sharing practices of its communication network. Myers (1994) examined the contribution of computer networks to the formation and functioning of social movements and collective behavior.

Another relevant area of research aims to understand the processes of activist computer use and the results for social movements. Bennet (2003) addressed the capacity of digital media to change the political game and how the Internet is implicated in the new global activism. The Internet doesn’t just reduce the costs of communications, but rather transcends the geographical and temporal barriers associated with other communication media. Bennet et al. (2014) focused on crowd peer production and how crowd-enabled networks are activated, structured, and maintained. Garrett (2006) surveyed how new information and communication technologies are changing the ways in which activists communicate, collaborate, and demonstrate. Massa (2016) discussed
the role and growing influence of online communities and how they support faster, cheaper, and more flexible organizing, but also identifies the lack of empirical studies and how they become agents of social change. Lacking from these studies is how participants in social movements can benefit from anonymous communication and how privacy tools can help protect their identity and circumvent censorship.

Anonymity technologies, such as the Tor Network, can provide critical benefits to the participant during social movements. The use of these tools during periods of civil unrest has not been studied from a quantitative perspective. While there has been extensive research relating to the contribution of computer networks and social media to activism, there has been almost no quantitative research on the role and contribution of anonymity networks to social movements. According to the Tor Project, dissidents are advised to use Tor to ensure their privacy and safety. Jardine (2016) asks why people use anonymity-granting technologies when surfing the Internet. The author claimed that people use online anonymity services to evade repressive regimes infringing their civil and political rights. The results of this study concluded that Tor usage is driven by political repression (e.g. avoiding surveillance, circumventing censorship) as well as highly liberal contexts (e.g. free and easy access to technology). Rady (2013) showed that anonymity networks can become terrains for government-population conflict as they enable citizens to overpower governments’ conventional control mechanisms over cyber-information exchanges. Sandberg (Sandberg, 2017) developed anomaly detection and similarity methods that used Tor client and Twitter usage data to detect country-level anomalous behavior and identify similar patterns across multiple countries. This research
provided novel techniques to detect both censorship and protest events and leveraged social media data to help explain the occurrence of events.

The goal of this study is to develop an automated protest event prediction tool using anonymous communication usage patterns and protest event data sources. A key component of maximizing prediction accuracy is proper selection of algorithms and configuration of their parameters. Machine learning algorithms take a long time to train and their hyperparameters must be configured prior to model training. Model configuration can be thought of as including the preprocessing methods, the classification methods, and the various boolean, categorical, discrete, and real-valued variables that configure these methods. Hyperparameters usually have a significant effect on the performance of the resulting models. In practice, manually tuning learning parameters is time consuming and prone to suboptimal solutions (Hutter, Lücke, & Schmidt-Thieme, 2015). The problem of finding the best configuration from all feasible configurations is a massive search optimization problem. The characteristics of search problems depends on the prediction function (e.g. neural network), the chosen error function (e.g. error rate), and training data. Complex models can overfit the data resulting in high variance. Models with too low complexity can underfit the data, failing to capture all the information in the data and result in high bias. Properly tuning model parameters and leveraging automated search strategies aims to address this bias-variance trade-off. Automating the search problem has been receiving increased attention in the machine learning community. A theoretically sound search strategy is essential to optimize the performance of any given learning algorithm for a given problem. A wide variety of strategies exist for exploring
and exploiting the search space including simple grid search, random search (J. Bergstra & Bengio, 2012; J. S. Bergstra, Bardenet, Bengio, & Kégl, 2011), gradient-based optimization (Bengio, 2000; Kingma & Ba, 2014), genetic algorithms (Olson & Moore, 2016; Tsai, Chou, & Liu, 2006), and sequential model-based Bayesian optimization (SMBO) methods (J. Bergstra, Komer, Eliasmith, Yamins, & Cox, 2015; Snoek, Larochelle, & Adams, 2012).

The algorithm configuration problem can be formally stated as follows: given a parameterized algorithm A, a set of problem instances I and a cost metric c, find parameter settings of A that minimize c on I (Hutter, Hoos, & Leyton-Brown, 2011).

While numerous optimization strategies exist for automating the algorithm configuration problem, all aim to find an optimized model as efficiently as possible. They all must deal with both low and high dimensionality and different evaluation budgets. While a comprehensive survey of strategies to search optimization in machine learning is beyond the scope of this research, several important contributions were reviewed for useful background information, including review of the search strategy selected and used in the analysis.

The SMBO class of optimization methods use a surrogate function to approximate a black-box function (i.e. the objective or cost function is unknown) and iterates between fitting models and using them to make choices about which configuration to investigate next. To date, it has been a popular strategy used to automate the configuration of machine learning algorithms. Approaches of this type include Bayesian optimization,
Sequential Model-based Algorithm Configuration (SMAC), and Tree-structured Parzen Estimators (TPE).

Bayesian optimization is a method of finding the maximum of expensive cost functions (Brochu, Cora, & de Freitas, 2010; Mockus, 2012). It can be used to model a learning algorithm’s generalization performance as a sample from a stochastic process such as a Gaussian Process (GP) (Snoek et al., 2012). GPs are a particular type of statistical model where observations occur in a continuous domain. They can be used to more flexibly fit nonlinear functions, similar to nonlinear regression. GP models provide a probabilistic approach to learning kernels and can be used in place of other models including Bayesian linear models, large neural networks, and support vector machines (a non-probabilistic approach) (Rasmussen, 2006). They are well known for being a good method for modeling loss functions in regression and classification tasks. In Bayesian optimization, GPs view tuning parameters as the optimization of the unknown black-box function and invoke algorithms developed for such problems. Bayesian optimization is a special case of nonlinear optimization where the algorithm decides which point to explore next based on the analysis of a distribution over the GP as the surrogate model. This approach models $P(y|x)$ directly, where $x$ represents hyperparameters and $y$ represents the associated quality score. Spearmint and the Bayesian optimization package are two implementations of this strategy and have shown suitable for low-dimensional numerical hyperparameter optimization. fn Spearmint was designed to automatically run experiments

---

1 Spearmint package https://github.com/HIPS/Spearmint; Bayesian optimization package https://github.com/fmfn/BayesianOptimization
in a manner that iteratively adjusts a number of parameters so as to minimize some objective in as few runs as possible.

SMAC is a more sophisticated approach to the general Bayesian optimization framework (Hutter et al., 2011). It has been used to speed up local search and tree search algorithms by orders of magnitude on certain instance distributions. It has been used to scale better to high dimensions and for discrete-based input dimensions. SMAC captures and exploits important information about the model domain, such as which input variables are most important. It uses a random forest of regression trees to model the objective function and has been shown to perform well on joint algorithm selection and hyperparameter optimization in deep neural networks.\(^2\) Auto-WEKA is an automated machine learning library based on the open source WEKA package and uses SMAC to perform combined algorithm selection and hyperparameter optimization (Kotthoff, Thornton, Hoos, Hutter, & Leyton-Brown, 2017; Thornton, Hutter, Hoos, & Leyton-Brown, 2012, 2013). SMAC was also used in the auto-sklearn automated machine learning toolkit (Feurer et al., 2015). Auto-sklearn is a re-implementation of Auto-WEKA, and includes meta-learning to bootstrap Bayesian optimization and automatic construction of ensembles.

TPE is another approach to sequential model-based Bayesian optimization and may provide some improvements over SMAC. TPE uses these two separate models to model the posterior. This approach models \(P(x|y)\) and \(P(y)\), where \(x\) represents

hyperparameters and $y$ the associated quality score. $P(x|y)$ is modeled by transforming the generative process of hyperparameters, replacing the distributions of the configuration prior with non-parametric densities (J. S. Bergstra et al., 2011; J. Bergstra, Yamins, & Cox, 2013). Hyperopt library provides an implementation of TPE for optimizing over search spaces with real-valued, discrete, and conditional dimensions (J. Bergstra et al., 2015).³ Hyperopt provides a parameterization of a search space over sequences of preprocessing steps and classifiers. It allows the user to choose the search domain, the objective function, and the optimization algorithm. The search domain is the set of valid assignments to the parameters of the learning algorithm. It includes operators and functions that combine random variables into data structures for the objective function. The objective function maps a joint sampling of these random variables to a scalar-valued score (e.g. cross-validation) that the optimization algorithm will try to minimize. The objective function performs model training and validation. The optimization algorithm performs the analysis of finding the best performing configuration and returns that result. From a usage perspective, the user creates an estimator object and invokes a fit function with training data and training labels. It uses a sequence of preprocessing steps and a classifier as if they were just one component. The fit function searches the space of preprocessing and classification steps along with the hyperparameters and returns an optimized model. The optimized model instance is then used on new data and scored for accuracy.

³ Hyperopt library https://github.com/hyperopt/hyperopt
The final strategy is based on an evolutionary computation technique called genetic programming. The approach, tree-based pipeline optimization, aims to automatically build programs to solve problems independent of the domain. It automatically constructs a series of data transformation and learning algorithms that maximize prediction accuracy (Olson et al., 2016). Machine learning operators are used as genetic programming primitives and the primitives are combined into working machine learning pipelines. Pipelines are represented using a tree-based structure and the genetic algorithm (GA) is used to evolve the pipelines. GA is well suited for solving optimization problems with many possible solutions. GA generates N (e.g. N=100) random tree-based pipelines and evaluates the cross-validation accuracy on the dataset. For each generation, GA selects the top M (e.g. M=20) pipelines in the population according to a selection scheme. Selection is based on the use of a multi-objective optimization technique called Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb, Pratap, Agarwal, & Meyarivan, 2002). The NSGA-II algorithm is used to select individuals based on two objectives, the pipeline length and performance. The final pipeline is selected based on maximum classification accuracy and minimum number of operators in the pipeline. The current implementation of the tree-based pipeline optimization approach could be improved with higher quality and faster pipeline construction. One approach to improve performance is to first evaluate against a small subset of the data and only evaluate the most promising pipelines on the full dataset (Gijsbers, Vanschoren, & Olson, 2018). This approach of allocating resources to more promising pipelines is also used in Hyperband (Li, Jamieson, DeSalvo, Rostamizadeh, &
Talwalkar, 2016). Hyperband can be thought of as a fast version of Random Search that adaptively allocates resources across the selected configurations. Tree-based pipeline optimization was selected to support the development of an automated approach to predicting Tor escalation during protest events.
DATA PREPARATION

The Armed Conflict Location and Event Dataset (ACLED) is a comprehensive daily collection of political and protest events that focus on the African continent\(^4\). The data represents a high quality, human vetted source of conflict events that captures the most accurate dates, location information, and related details. At the time of this writing, ACLED only provided data for Africa, but they have plans to provide similar data for the Middle East and Asia. A variety of factors contributed to the selection of the nine African countries used in the analysis. The countries were selected based on large populations, high Internet penetration rates\(^5\), and the degree of freedom based on the Freedom House Index\(^6\). An even distribution of “free”, “partly free”, and “not free” countries were used in the study. Freedom indicators supported the determination of whether there existed a stronger signal of Tor usage in authoritarian regimes. Countries were also vetted for having an active history of protest events and that the events occurred over the time period of the study.

Africa is categorized into five regions including Eastern, Central, Northern, Southern, and Western. All regions except the Central region were represented in the analysis. The Central region lacked countries with appropriate levels of Internet penetration to make them viable for use in this analysis. Countries were only included if

---

\(^4\) Event data was collected from the ACLED website [https://www.acleddata.com/](https://www.acleddata.com/)
\(^5\) Population and Internet statistics were collected from Internet World Stats website [https://www.internetworldstats.com/stats1.htm](https://www.internetworldstats.com/stats1.htm)
\(^6\) Indicators for a countries degree of freedom were collected by the Freedom House organization website [https://freedomhouse.org/](https://freedomhouse.org/)
they had at least a twenty-five percent Internet penetration rate to ensure that the target variable (i.e. Tor escalation) for the classification task provided an adequate concentration of Internet users. Three countries from each freedom index rating were selected. Table 1 shows the nine countries that met all criteria for inclusion. The time period for the data used in the analysis covered September 1, 2011 through December 31, 2016.

Table 1. Selected Countries.
The selected countries are ranked by Internet penetration rate (%).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Kenya (ke)</td>
<td>Eastern</td>
<td>48,466,928</td>
<td>37,718,650</td>
<td>77.8%</td>
<td>Partly Free</td>
</tr>
<tr>
<td>Morocco (ma)</td>
<td>Northern</td>
<td>35,241,418</td>
<td>20,207,154</td>
<td>57.3%</td>
<td>Partly Free</td>
</tr>
<tr>
<td>South Africa (za)</td>
<td>Southern</td>
<td>55,436,360</td>
<td>28,580,290</td>
<td>51.6%</td>
<td>Free</td>
</tr>
<tr>
<td>Tunisia (tn)</td>
<td>Northern</td>
<td>11,494,760</td>
<td>5,800,000</td>
<td>50.5%</td>
<td>Free</td>
</tr>
<tr>
<td>Nigeria (ng)</td>
<td>Western</td>
<td>191,835,936</td>
<td>93,591,174</td>
<td>48.8%</td>
<td>Partly Free</td>
</tr>
<tr>
<td>Libya (ly)</td>
<td>Northern</td>
<td>6,408,742</td>
<td>2,800,000</td>
<td>43.7%</td>
<td>Not Free</td>
</tr>
<tr>
<td>Algeria (dz)</td>
<td>Northern</td>
<td>41,063,753</td>
<td>15,105,000</td>
<td>36.8%</td>
<td>Not Free</td>
</tr>
<tr>
<td>Egypt (eg)</td>
<td>Northern</td>
<td>95,215,102</td>
<td>34,800,000</td>
<td>36.5%</td>
<td>Not Free</td>
</tr>
<tr>
<td>Ghana (gh)</td>
<td>Western</td>
<td>28,656,723</td>
<td>7,958,675</td>
<td>27.8%</td>
<td>Free</td>
</tr>
</tbody>
</table>

The ACLED database provides detailed codes for the specific type of event, the source and target actors involved in the event, and each event was recorded with date and geographic coordinates (Raleigh et al., 2010). The information was highly curated and
manually coded providing a reliable, thorough, and usable source of conflict event data.

The analysis used the nine different ACLED event types and eight actor categories as shown in Table 2. Actor categories were included for both the source actor (perpetrator) and the target actor (victim).

Table 2. ACLED Event Types and Actor Categories.
The chart shows the total number of events for each country with the percentage of events coded as Riots or Protests (7). The median daily Tor usage counts are also shown per country. One or more events can occur on the same day. In most cases, an event involves two actors, the source actor and target actor. Single actor events are recorded with No Actor (0) for the target actor. Event and Actor codes are shown in the notes.

<table>
<thead>
<tr>
<th>Country</th>
<th>Total Events</th>
<th>Protest Events (%)</th>
<th>Tor Usage (Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Egypt</td>
<td>6900</td>
<td>3801 (.55)</td>
<td>4581</td>
</tr>
<tr>
<td>Nigeria</td>
<td>6766</td>
<td>2438 (.36)</td>
<td>1541</td>
</tr>
<tr>
<td>South Africa</td>
<td>6480</td>
<td>5433 (.84)</td>
<td>5688</td>
</tr>
<tr>
<td>Libya</td>
<td>4481</td>
<td>789 (.18)</td>
<td>681</td>
</tr>
<tr>
<td>Tunisia</td>
<td>2548</td>
<td>1962 (.77)</td>
<td>2229</td>
</tr>
<tr>
<td>Kenya</td>
<td>2195</td>
<td>955 (.44)</td>
<td>927</td>
</tr>
<tr>
<td>Algeria</td>
<td>1707</td>
<td>1049 (.61)</td>
<td>2833</td>
</tr>
<tr>
<td>Morocco</td>
<td>708</td>
<td>512 (.72)</td>
<td>3026</td>
</tr>
<tr>
<td>Ghana</td>
<td>368</td>
<td>264 (.72)</td>
<td>759</td>
</tr>
</tbody>
</table>

Notes:

**EVENT TYPES**
1 - Battle-Government regains territory
2 - Battle-No change of territory
3 - Battle-Non-state actor overtakes territory
4 - Headquarters or base established
5 - Non-violent transfer of territory
6 - Remote violence
7 - Riots/Protests
8 - Strategic development
9 - Violence against civilians

**ACTOR CATEGORIES**
0 - No Actor
1 - Government
2 - Rebel force
3 - Political militia
4 - Ethnic militia
5 - Rioters
6 - Protesters
7 - Civilians
8 - Outside force
The number of events recorded in ACLED can vary from zero to dozens on any given day. For analysis purposes, an equally measured time series dataset was created with the temporal unit of a single day. The data was prepared to minimize any loss of information. To accomplish this, the discrete categorical event and actor features were numerically coded, and then binary encoded with a new boolean feature for each event type and actor category. Only one of the features takes on the value 1 for each sample and the transformed data becomes a higher dimensional sparse matrix. This method is typically referred to as one-hot encoding and is performed to improve a dataset for machine learning. Because there can be multiple events occurring on a single day, a single record per day was created by summing the event types and actor categories to create a count value for each of these features. Days without any events occurring were recorded with zeros for all features. This approach collapsed the raw ACLED data into a single instance per day with a count of the total number of events, a count for each event type, a count for each source and target category, and a count of fatalities. Together, these event features provide a representation of conflict behavior for each day in each country.

The next step was to prepare the target variable based on Tor usage.

The Tor network is the most mature and largest deployed anonymous communication network that conceals the user’s location, identity, and activity. The entire public Tor network includes over 7,000 relay nodes with varying levels of bandwidth. Relays and directory authorities publish relay descriptors so that clients can select relays for their paths through the Tor network. Tor relay nodes perform data
collection services throughout the Tor network. Tor usage metrics are collected by the Tor relay nodes and country-level aggregate data is provided by the Tor Project for the research community\textsuperscript{7}. There are several data layers provided by the Tor network to support client usage metrics, including the relay server descriptors, relay extra-info descriptors, and network status consensus.

Counting client users in the Tor network is somewhat challenging since collecting client information could reduce the anonymity and safety of the user. Therefore, the data on Tor usage metrics are only available at the aggregate level. Country-level usage statistics provide client counts representing the number of unique users connecting to the Tor network on a daily basis.

Client usage counts for each of the selected countries were extracted from the Tor user metrics. New rows were created for each missing day and the missing values were filled with the previous day’s known value. Only four days of Tor usage data was missing over the entire study period. Tor escalation models were developed to measure whether Tor usage was increasing or decreasing on a daily basis. Two escalation methods were developed including a binary escalation model, where a 1 indicates an increase in usage or no more than a 2% decrease from the previous day, and a 0 indicates a decrease of more that 2% from the previous day. The second method used percent change with a split assignment giving all percent changes below the mid-quartile a 0 and all values above mid-quartile a 1. Class labels (target variables) were created for each of these escalation

\textsuperscript{7} Tor usage metrics were provided by the Tor Project \url{https://www.torproject.org/} and Tor Metrics \url{https://metrics.torproject.org/}.
models plus additional class labels that captured time lags of 1 and 2 days in both directions. This resulted in a total of ten class labels that could be used by the classification methods described in the Methodology section. The set of targets provide a 5-day window for analyzing temporal behaviors in Tor usage in relation to protest events. They capture behaviors that may be occurring prior to an event or behaviors in response to an event.

Visualizing data helps to understand the relationship between ACLED events and Tor usage. Time series plots are useful for visually displaying this relationship. Figure 1 shows the time series plots for the entire study period for all nine African countries.
Figure 1. Time Series Plots for ACLED Events and Tor Usage.
The plots show the relationship between ACLED events and Tor usage for each country. The plots are grouped by free (Tunisia, Ghana, South Africa), partly-free (Morocco, Nigeria, Kenya), and not free (Egypt, Libya, Algeria) countries.

It can be difficult to see clear patterns while visualizing the full time period of the study. It is often helpful to visualize sample periods such as those shown in Figure 2. The plot on the left shows a random sample from the South Africa dataset covering a two-month period of September 1, 2016 through October 31, 2016. The plot on the right shows the behaviors exhibited during the 2013 Egyptian protests, which took place over a three-day period of June 30, 2013 through July 3, 2013. The goal of this protest was to overthrow Mohamed Morsi and suspend the constitution. The sample periods reveal a markedly similar temporal pattern between ACLED events and Tor usage patterns. This study aims to determine if this type of pattern generally exists for the different countries and if it is possible to accurately predict Tor usage during protest events.
The last step in data preparation process included finding spatial clusters of events and using the resulting cluster labels as new features characterizing events. Spatial clustering was used to find spatial groupings for the event data on a per country basis, with the number of clusters represented by the variable K. The K-means algorithm was used to iteratively assign each data point to one of K groups based on the geographic coordinates provided for each event (Arthur & Vassilvitskii, 2007).

The user of the K-means algorithm is required to select the value for K, which is the number of clusters to group the data. This can be difficult to do since there is no easy way to select the best K. One option is to use metrics on the clusters to automatically find the best value for K. The silhouette coefficient measures mean intra-cluster distance (m) and mean nearest-cluster distance (n) for each sample and then calculates a single metric using the formula $(n - m) / \max(m, n)$ (Rousseeuw, 1987). The silhouette defines how well clusters are fit by the model and ranges from -1 (poorly matched to neighboring
clusters) to 1 (well matched to its own cluster). This metric was used to iteratively find the optimal value for K when fitting the K-means clustering algorithm to each country dataset.

Map visualizations help to understand the resulting clusters, which include the geographic extent and spatial relationships among conflict events. Again, a sample is used to visualize resulting clustering behaviors. In Figure 3, the map shows the spatial patterns and grouping of events into clusters for South Africa. The map displays five spatial clusters with the distribution of events per cluster. An additional overlay shows the distribution of the events that occurred in South Africa over the sample time period from Figure 2. This sample distribution is spread across all clusters indicating that this particular slice into the data does not have a spatial-temporal pattern consistent with a particular geographic cluster.
The events tend to cluster around the population of the nine provinces of South Africa. The largest cluster covers the provinces of Northern Cape, North West, Free State, and Gauteng (2211 events). The other clusters align with the remaining provinces of Limpopo and Mpumalanga (500 events), KwaZulu-Natal (1161 events), Eastern Cape (1044 events), and Western Cape (1564 events).

New features were derived from the spatial clusters and used to determine if spatial information can improve prediction accuracy of the classifiers. The labels resulting from the spatial clustering were added to each event in the datasets, applying the same one hot encoding process used for event types and actor categories described above. Figure 4 displays the entire set of features as a sparse matrix time series. The features include the total number of events that occurred each day, along with the count of the event types, source actors, target actors, fatalities, and cluster labels.
Figure 4. Input Features for the Classification Tasks.

Each feature value is represented as a sum, as multiple events can occur on any given day. The class label indicates whether Tor escalated that day (1) or did not escalate (0).
METHODOLOGY

The goal of supervised machine learning is to make predictions as accurately as possible. A variety of classification methods were selected, implemented, and evaluated to predict the escalation in Tor usage during protests. If models produce accurate scores, then it is reasonable to assume that a relationship exists between events and Tor usage. The approach discovered models that work best for the problem domain and the different datasets. Two analytic workflows were explored to achieve this goal. The first workflow was to manually explore, configure, and evaluate different classification models and then choose the best performing model. The second workflow, which is described in the next section, was to leverage automated machine learning to automatically discover the optimal solutions. The manual approach provided baseline solutions to compare against the automated approach. The goal was to determine if an applied approach outperforms an automated approach to predictive modeling and if the automated results, even if not as accurate, outweigh the time and effort required to manually produce good models.

Predictive Modeling

Twelve different classification methods were evaluated to predict the escalation in Tor usage during political events. This approach resulted in the selection of a classification method that produced the maximum predictive power for each of the nine African country datasets. Each classification method was fit to each of the nine event datasets and iteratively processed against the labeled test data. The algorithms were configured by tuning the hyperparameters or using the recommended default settings.
The classifiers evaluated included linear, non-linear, ensembles, and deep leaning estimators. Each classification method uses a specific strategy to determine the decision boundaries given the variances in the datasets. It was assumed that some datasets would be more easily separated linearly, while other datasets would require a non-linear or an ensemble approach to produce an optimal solution.

The twenty-eight data features described in the Data Preparation section captured the unique characteristics of the events for each country. A separate experiment was run that incorporated the cluster labels to determine if this spatial information would improve performance. The input features remained constant across all model iterations. A summary of the methodology is shown in Figure 5.
Figure 5. Predictive Modeling Workflow.
The approach allows for a variety of experiments to be conducted easily. New classifiers can be added and different combination of input features can be evaluated.

While certain algorithms are better suited for particular problems, certain assumptions and constraints must be considered when selecting an algorithm. For instance, some algorithms can work with categorical features or will automatically convert them to numerical values, while other algorithms fail with this data type. For this reason, all categorical features were one-hot encoded to produce an all numeric dataset. Another common consideration is handling missing data. Some algorithms fail with missing data and require missing data imputation to occur prior to learning, while other learners may be resilient to missing data or automatically impute values. Therefore, the datasets were prepared and held constant across all evaluations to ensure all methods would execute successfully.

Algorithm Selection and Hyperparameter Tuning
Algorithm selection for a problem depends on multiple factors including number of samples and features in the given dataset, the data types and distributions of those features, and the independence or correlations between features and the target variable. The process involved development and exploration of the different classification models using a variety of configurations for each model to optimize its performance. The deeper the understanding of each algorithm can help reduce the amount of time to configure each one properly. Understanding the underlying behavior of the different learning algorithms
on each dataset is a challenging undertaking. For each model, parameters were manually tuned via trial and error to achieve optimal performance on each dataset. The following methods and hyperparameter settings were implemented. The parameters shown are a subset of available parameters, with the remaining parameters assigned to their default values.

K-Nearest Neighbors classifier is an instance-based learning method that stores instances of the training data and does not construct a general internal model. Classification is based on majority vote using the k-nearest neighbors at each instance. The choice for parameter k (n_neighbors) is dependent on the data. With a larger k, the effects of noise are reduced, but the boundaries become less distinct. The weights parameter is used to assign uniform weight (all points in each neighborhood are weighted equally) or distance weight (closer neighbors to an instance will have a greater influence than neighbors further away). Because the dataset in this study was relatively small, a brute force algorithm and Euclidean distance (p=1) were used to compute the nearest neighbors. The parameters for K-Nearest Neighbors were assigned as follows: (n_neighbors=3, weights='uniform', algorithm='brute', p=1).

Decision Tree classifier creates models by learning decision rules (a set of if-then-else rules) that are inferred using the data features. Decision trees are easy to understand and interpret because the trees can be visualized. This is referred to as a white box model versus a black box model where results are difficult to interpret (e.g. neural networks). Decision trees support both numeric and categorical data and can handle both binary and multi-output prediction problems. A simple method that performs well and is
interpretable is sometimes preferred over complex models that only improve accuracy slightly. Key disadvantage of decision trees is that they are prone to create complex trees that overfit the training data. Approaches to minimize overfitting include setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree. To deal with the unstable nature of decision trees (i.e. a small variation in the data resulting in significantly different trees) it is possible to use decision trees in ensemble methods (e.g. Random Forest Classifier). The parameters for Decision Tree were assigned as follows: (criterion=’entropy’, max_depth=3, min_samples_leaf=2).

Random Forest classifier is an ensemble method that constructs multiple decision trees on various sub-samples of the data. Ensemble learning combines several models that learn and make predictions independently, which are then combined into a single prediction. Random Forest uses a mean across decision trees to maximize accuracy and minimize overfitting that can occur with a single decision tree. During training, this method applies a bagging or bootstrap aggregation technique in tandem with random feature selection. Bagging is used to enhance accuracy when random features are used and to give ongoing estimates of the generalization error of the combined ensemble of trees (Breiman, 2001). This method is also good at feature selection or importance since it is possible to examine which features are working best in each tree. While not as difficult to configure as Support Vector Machines (SVM), there are still many options for experimentation. The n_estimators is the number of trees to use in the forest. Criterion is the function used to measure the quality of the split and can be assigned Gini impurity or entropy for the information gain. The max_features is the number of features to consider.
when looking for the best split. The max_depth is the maximum depth of the tree. The parameters for Random Forest were assigned as follows: (n_estimators=10, criterion='entropy', max_features=3, max_depth=5).

AdaBoost classifier is a boosting algorithm that fits a sequence of weak learners (i.e. better than random guessing) on repeatedly modified versions of the data. The predictions are then combined through a weighted majority vote to produce final prediction. The number of weak learners is controlled by n_estimators and the learning_rate controls the contribution of the weak learners in the final combination. AdaBoost can be used in conjunction with other learning methods to improve performance. The base_estimator is set to the default decision tree classifier as the week learners. The algorithm is set to the SAMME.R real boosting algorithm. The parameters for AdaBoost were assigned as follows: (DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_leaf=2), n_estimators=10, learning_rate=.01, algorithm='SAMME.R').

Support Vector Machine (SVM) classifier is a kernel machine based method that maps input features into a high-dimension feature space (Cortes & Vapnik, 1995). In this feature space a linear decision surface is constructed, which results in good generalization of the predictor. They can support both dense and sparse sample vectors as input. One of the disadvantages of SVMs is that they do not provide probability estimates. Probability estimates can be obtained using methods such as Gaussian Process. For SVM, probabilities can be calculated using cross-validation. SVMs use a subset of training points in the decision function (support vectors), so it is memory efficient. SVM can
produce acceptable models and predict target values with good accuracy when modeling sparse sample vectors as input, especially when the SVM is fit on sparse data (Boser, Guyon, & Vapnik, 1992). Several parameters of the SVM classifier were tuned. The parameter C is the penalty parameter of the error term. The kernel parameter specifies the different kernel functions that can be specified for the decision function. The different kernels include linear, polynomial, radial basis function (rbf), and sigmoid. The gamma parameter is the kernel coefficient. The parameters for Support Vector Machine were assigned as follows: (C=0.25, kernel='rbf', gamma=0.2). A linear SVM was also implemented with parameters assigned as follows: (C=0.1, kernel='linear').

Logistic Regression classifier is a linear model for classification where the probabilities describing the possible outcomes of a single instance are modeled using a logistic function. It implements regularized logistic regression using one of the following solvers: liblinear, newton-cg, sag, or lbfgs, and can handle sparse or dense data. In this implementation, a binary estimator was fit with an L2 regularization. Regularization is one way to avoid overfitting a model on training data. It does this by adding a penalty as model complexity increases and favors smooth functions. The regularization parameter (penalty) penalizes all parameters except the intercept so that the model generalizes the data and therefore does not overfit. Penalty can be the L1 regularization (Lasso regression) or L2 regularization (Ridge regression). The difference is based on how each assigns penalty to the coefficients. Ridge adds penalty equivalent to square of the magnitude of coefficients. Lasso adds penalty equivalent to absolute value of the magnitude of coefficients. Lasso may work well for feature selection where there are a
large set of features. Parameters set include tolerance for stopping criteria (tol), inverse of regularization strength (C), and solver for optimization. The parameters for Logistic Regression were assigned as follows: (penalty='l2', tol=1.0, C=0.01, solver='liblinear', multi_class='ovr').

Multi-layer Perceptron (MLP) classifier learns a non-linear function approximator. It is considered a black box model because it is considered difficult to interpret. It is the only deep neural network learning architecture included in the study. Neural networks try to mimic how a human brain learns. MLP optimizes (minimizes) the loss function using Limited-memory BFGS, stochastic gradient descent (SGD), or Adam. It trains using backpropagation, which is a method to calculate the error contribution of each neuron after a batch of data is processed (i.e. finds the local minimum of the error function). The training uses some form of gradient descent and the gradients are calculated using backpropagation. In other words, if the output generated is far away from the expected value, the weights and biases of the neurons are updated. Unlike logistic regression, there can be one or more non-linear hidden layers between the input layers and the output layer. The input layer represents the features (set of neurons). It uses a non-linear activation function (identity, logistic, tanh, or relu) for the hidden layer. The activation function performs a non-linear transformation over the input layer to the hidden layers. They decide whether a neuron should be activated or not. This helps the network take advantage of relevant information and filter irrelevant information. The output layer receives values from the last hidden layer and transforms them into output values. MLP requires significant tuning of hyperparameters including architecture, the
number of hidden neurons and layers, the maximum number of iterations (max_iter), the
activation function for the hidden layer, the solver for weight optimization, the L2
penalty parameter (alpha), the size of minibatches for stochastic optimizers (batch_size),
the learning rate schedule for weight updates, and many others. The parameters for Multi-
layer Perceptron were assigned as follows: (hidden_layer_sizes=(50,),
activation=’logistic’, solver=’adam’, alpha=1.0, max_iter=100,
learning_rate=’adaptive’).

The final two types of models are based on probabilistic classifiers and are
implemented using the default parameter settings. The first type is the Gaussian Process
(GP) classification approach, which is based on placing a prior distribution over the space
of functions rather than on parameters. GPs have been frequently used in statistics, geo-
statistics, time series analysis, and as spatial models in meteorology and geology. GPs are
closely related to support vector machines (SVM). They can be interpreted as a Bayesian
version of SVM, using a probabilistic approach to kernel-based learning (Rasmussen,
2006). They are easier to work with than SVMs and are easier to interpret model
predictions as compared to neural networks. The implementation is based on Laplace
approximation for the binary classifier, which is used for approximating the non-
Gaussian posterior by a Gaussian. It uses an RBF kernel with default parameters. The
second type is based on Naïve Bayes, which uses conditional probability (i.e. the
probability something will happen given that something else has already occurred). These
methods are popular for binary classification in spam detection, document classification,
and medical diagnosis. The Gaussian Naïve Bayes classifier assumes features are
continuous values, all features are independent and follow a Gaussian distribution, and each feature contributes independently to the probabilistic outcome of an event. The MultinomialNB and BernoulliNB classifiers are variants and use the multinomial distribution and multivariate Bernoulli distributions, respectively. The Bernoulli classifier requires samples to be represented as binary-valued feature vectors, which is consistent with most of the data used in this study. Table 3 summarizes the models and hyperparameter settings.

Table 3. Summary of Classification Models and Model Hyperparameters.

<table>
<thead>
<tr>
<th>Model</th>
<th>Model Hyperparameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-Nearest Neighbors</td>
<td>(n_neighbors=3, weights='uniform', algorithm='brute', p=1)</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>(criterion='entropy', max_depth=3, min_samples_leaf=2)</td>
</tr>
<tr>
<td>Random Forest</td>
<td>(n_estimators=10, criterion='entropy', max_features=3, max_depth=5)</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>(DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_leaf=2), n_estimators=10, learning_rate=.01, algorithm='SAMME.R')</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>(C=0.25, kernel='rbf', gamma=0.2)</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>(C=0.1, kernel='linear')</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>(penalty='l2', tol=1.0, C=0.01, solver='liblinear', multi_class='ovr')</td>
</tr>
<tr>
<td>Multi-Layer Perceptron</td>
<td>(hidden_layer_sizes=(50,), activation='logistic', solver='adam', alpha=1.0, max_iter=100, learning_rate='adaptive')</td>
</tr>
<tr>
<td>Gaussian Process</td>
<td>default parameters</td>
</tr>
<tr>
<td>GaussianNB</td>
<td>default parameters</td>
</tr>
<tr>
<td>MultinomialNB</td>
<td>default parameters</td>
</tr>
<tr>
<td>BernoulliNB</td>
<td>default parameters</td>
</tr>
</tbody>
</table>
Automated Machine Learning

Different strategies to automating machine learning pipelines address performance, scalability, and optimization in unique ways. The automated search process focuses on quality and efficiency in finding a good model and their configuration in the large space of potential models and configurations. In general, the search process takes as input a training dataset, a description of the model space to search, and a budget or stopping criterion. The model space includes the various algorithms (e.g. neural network, decision tree, logistic regression, SVM, random forest) and associated configuration parameters. The budget can be defined in a variety of ways including total number of models to train, total number of model configurations, maximum execution time, or compute cost. For classification problems, the output of these tools is a model with the best prediction accuracy.

The automated approach utilized the tree-based pipeline optimization method (Olson & Moore, 2016) to automatically construct machine learning pipelines. Thousands of model pipelines were designed and optimized using the ACLED crisis event and Tor usage datasets. The predictive modeling workflow was significantly reduced limiting the development to data preparation and executing the automated machine learning module as shown in Figure 6.
Automated machine learning tools currently have limited capabilities for automated data cleaning and other data preparation methods. The data was prepared exactly the same as was done for the manual process.

The search component used genetic programming (GP) to automatically construct and optimize the pipelines. Machine learning algorithms were used as GP primitives and the GP algorithm was used to combine primitives and evolve the tree-based pipelines into models. The search process is stochastic so resulting pipelines can be different across each run. The classification methods searched included: GaussianNB, BernoulliNB, MultinomialNB, DecisionTreer, ExtraTreesr, RandomForest, GradientBoosting, KNeighbors, LinearSVC, and LogisticRegression. In addition, a variety of preprocessors, decomposition, kernel approximation, clustering, and feature selection primitives were also searched including: Binarizer, FastICA, FeatureAgglomeration, MaxAbsScaler,
MinMaxScaler, Normalizer, Nystroem, PCA, PolynomialFeatures, RBFSampler, RobustScaler, StandardScaler, ZeroCount, OneHotEncoder, SelectFwe, SelectPercentile, VarianceThreshold, RFE, and SelectFromModel.

Configuration of the GP-based search component involved setting the number of generations (G), the population size (P), and the offspring size (O). G represents the number of iterations to run the pipeline optimization process. P represents the number of individuals to retain in the GP population every generation. O represents the number of offspring to produce in each GP generation. P + G * O determines the number of pipeline configurations that will be evaluated. For the experiments conducted, G was assigned 5, P was assigned 20, and O was assigned 20. Since 5 different target variables were evaluated and 5-fold cross validation was used, this resulted in ((P + G * O) * 5 * 5) or 3,000 pipeline configurations fit and evaluated for each dataset. The fit function initializes the GP algorithm to find the highest scoring model based on the mean five-fold cross validation score. This involved significantly more models being searched and evaluated compared to the twelve models evaluated in the manual approach.
RESULTS

For each evaluation of the nine datasets, the data was split into training and testing data using a 90/10 split and run across all models identified in Table 3. Results were compared using prediction test accuracy on held out test data (mean accuracy on the given test data and target variable). Test accuracy was also measured using the F1 score metric, which is commonly used for binary classification problems and considers both precision and recall to compute the score:

\[
F1 \text{ Score} = 2 \frac{(Precision \times Recall)}{(Precision + Recall)},
\]

\[
Precision = \frac{True \ Positives}{(True \ Positives + False \ Positives)},
\]

\[
Recall = \frac{True \ Positives}{(True \ Positives + False \ Negatives)}.
\]

To assess how well a model will generalize to unseen data, cross validation was used and the mean cross validation score was reported in the results. A five-fold cross validation technique was used for each evaluation. Finally, a measure to quantify uncertainty was reported to help determine how much confidence a stakeholder should have in the performance of the trained models. Uncertainty of model accuracy was computed using the 95% confidence interval. The results of the predictive modeling approach are shown in Table 4, with the classifier and target variable shown along with the best overall accuracy scores for each country.
The models produced reasonably accurate scores, so it can be assumed that a relationship does exist between conflict events and Tor escalation. Quantitative evidence now exists for the use of Tor escalating prior to a conflict event or responding to an event as demonstrated in the different time lags. Countries that are rated as not-free all have the highest accuracy scores and are highlighted in red. This indicates that these countries have the highest signal in the use of Tor in relation to conflict events as compared to those rated as free or partly-free. Participants in protests taking place in countries with an authoritarian regime may feel the need to protect their identity and location or need to circumvent censorship during these conflict periods.
The results of the automated machine learning approach are shown in Table 5, with the best model, target variable, and model hyperparameters shown along with the best overall accuracy score and cross validation score for each country. The same 90/10 training and testing split was used along with five-fold cross validation.

Table 5. Automated Machine Learning Results.
While the automated model selections were different than manually selected models, the accuracy of the different models are comparable. Again, the most accurate models related to countries with authoritarian regimes (highlighted in red). CV – Cross Validation.

<table>
<thead>
<tr>
<th>Country</th>
<th>Best Model</th>
<th>Best Target</th>
<th>Model Parameters</th>
<th>Accuracy</th>
<th>CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Egypt</td>
<td>Random Forest</td>
<td>Time Lag -1</td>
<td>(bootstrap=False, criterion=&quot;entropy&quot;, max_features=0.05, min_samples_leaf=16, min_samples_split=20, n_estimators=100)</td>
<td>0.764</td>
<td>0.698</td>
</tr>
<tr>
<td>Algeria</td>
<td>Logistic Regression</td>
<td>Time Lag 0</td>
<td>(PolynomialFeatures(degree=2, include_bias=False, interaction_only=False), LogisticRegression(C=0.1, dual=False, penalty=&quot;l1&quot;))</td>
<td>0.749</td>
<td>0.69</td>
</tr>
<tr>
<td>Libya</td>
<td>Logistic Regression</td>
<td>Time Lag -1</td>
<td>(C=0.1, dual=False, penalty=&quot;l2&quot;)</td>
<td>0.744</td>
<td>0.686</td>
</tr>
<tr>
<td>Tunisia</td>
<td>Decision Tree</td>
<td>Time Lag -2</td>
<td>(FeatureAgglomeration(affinity=&quot;precomputed&quot;, linkage=&quot;complete&quot;), DecisionTreeClassifier(criterion= &quot;entropy&quot;, max_depth=2, min_samples_leaf=17, min_samples_split=11))</td>
<td>0.738</td>
<td>0.678</td>
</tr>
<tr>
<td>Country</td>
<td>Algorithm</td>
<td>Time Lag</td>
<td>Parameters</td>
<td>ACC</td>
<td>MCSI</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>Morocco</td>
<td>Random Forest</td>
<td>+2</td>
<td>(bootstrap=False, criterion=&quot;entropy&quot;, max_features=0.05, min_samples_leaf=16, min_samples_split=20, n_estimators=100)</td>
<td>0.713</td>
<td>0.655</td>
</tr>
<tr>
<td>South Africa</td>
<td>Decision Tree</td>
<td>-2</td>
<td>(criterion=&quot;entropy&quot;, max_depth=2, min_samples_leaf=17, min_samples_split=7) (Binarizer(threshold=1.0),</td>
<td>0.641</td>
<td>0.634</td>
</tr>
<tr>
<td>Kenya</td>
<td>Decision Tree</td>
<td>-2</td>
<td>DecisionTreeClassifier(criterion=&quot;entropy&quot;, max_depth=2, min_samples_leaf=17, min_samples_split=4)</td>
<td>0.621</td>
<td>0.572</td>
</tr>
<tr>
<td>Ghana</td>
<td>Random Forest</td>
<td>-2</td>
<td>(bootstrap=False, criterion=&quot;entropy&quot;, max_features=0.05, min_samples_leaf=2, min_samples_split=20, n_estimators=100)) (RFE(estimator=ExtraTreesClassifier(criterion=&quot;gini&quot;, max_features=1.0, n_estimators=100), step=0.8), SelectPercentile(score_func=f_classif, percentile=18), LinearSVC(C=25.0, dual=True, loss=&quot;squared_hinge&quot;, penalty=&quot;l2&quot;, tol=0.1))</td>
<td>0.615</td>
<td>0.584</td>
</tr>
<tr>
<td>Nigeria</td>
<td>LinerSVC</td>
<td>+1</td>
<td>(RFE(estimator=ExtraTreesClassifier(criterion=&quot;gini&quot;, max_features=1.0, n_estimators=100), step=0.8), SelectPercentile(score_func=f_classif, percentile=18), LinearSVC(C=25.0, dual=True, loss=&quot;squared_hinge&quot;, penalty=&quot;l2&quot;, tol=0.1))</td>
<td>0.61</td>
<td>0.583</td>
</tr>
</tbody>
</table>

The automated approach generated comparable performance results to the manual predictive modeling process. While the automated approach experienced some degradation in predictive accuracy, no domain knowledge or expertise with the various machine learning algorithms were required. Increasing the number of generations, population size, and offspring size could improve performance, though evaluations would...
take much longer to complete. The automated method clearly produced high-quality accuracy results with the ability to fine tune the best model generated. Therefore, it is a good decision to apply automated machine learning to discover good models that otherwise may not be considered.
DISCUSSION

It is challenging to quantitatively account for the reason and degree of usage of the Tor Network. The practical uses of anonymity networks are diverse and include identity and location protection, surveillance avoidance, censorship circumvention, and criminal activities. While it is not possible to accurately measure how Tor clients are being used at any particular time, this research successfully detected signals that consistently demonstrated utilization of Tor during periods of civil unrest.

Tor usage patterns for the different countries can vary widely. Therefore, it was necessary to analyze a 5-day window in relation to conflict events so that the different temporal usage patterns would be adequately captured. This approach takes into account whether Tor usage escalates prior to, during, or after onset of a political event. Different Tor escalation models were evaluated and the binary escalation model was used, creating a total of five class labels. The class labels represent a time lag of one or two days prior to or after an event. Experiments were run using each class label against twelve classification methods and the one that produced the best prediction accuracy was selected for that country. If a classification method produces reasonably accurate scores, then it can be assumed that a relationship exists between political events and Tor usage escalation. Different escalation models can be developed that tighten or relax the definition of escalation.

One criterion for selection of countries to include in the study was based on Internet penetration rates. Proxy variable could also be explored to better understand the
use of online technologies in target countries, such as quantity of user-generated content on social media platforms, edits on Wikipedia, or the number of active Tor relays or bridges in a country. These proxy variables can help characterize the propensity for users in specific countries to leverage online tools. It should be noted that the absence of Tor relays or bridges in specific countries, such as in Libya, is likely due to the inherent risk to individuals hosting Tor infrastructure and is not a good measure of Tor usage. There is no requirement to have a Tor relay in the host country for users of that country to access the Tor network. Finally, analysis of content on specific Tor Hidden Services, can estimate usage of Tor for specific purposes. Hidden forums, blogs, chat rooms, and social networks that discuss civil unrest and protest events in target countries would provide an estimate on the number of accounts using Tor for specific purposes. These proxy variables and related analyses offer potentially interesting areas to extend this research.

Each country in the study has its own unique geography, demographic, government, and other factors that can shape political events or public discontent. Qualities of governance, degree of freedom, and access to Internet technologies have a direct impact on the patterns of collective action as well as the resulting behaviors in both physical and virtual spaces. These factors are intrinsically represented in the data and therefore create unique problem sets for each country. While spatial information can be used to explore the different temporal and spatial patterns of conflict events, the addition of spatial information provided using the spatial clustering labels as new features did not produce better performing models. Spatial analysis focusing on specific regions in
countries and specific event types or actor categories also offer potentially interesting areas to extend this research.

The methodology used in this study aimed to be consistent across all the datasets and to avoid overfitting through extensive feature selection or using specific slices of the input data. All evaluations used the same exact set of input features and used the full dataset for the entire study period for each country prior to fitting the models. Model improvements may be accomplished by removing anomalous periods in the Tor usage data. Additional approaches to consider include using multi-class classification rather than binary classification or conduct regression analysis on continuous changes in Tor usage versus the binary escalation model. Additional countries from the ACLED dataset could be added to the analysis including Sudan, Rwanda, Zimbabwe, and Uganda. Each of these countries have Internet penetration rates above 25%. This study ensured an even distribution of countries across the Freedom House index of free, partly-free, and not-free. These four countries are not-free, except Uganda, which is partly-free. Expanding the study beyond Africa would also be of interest, assuming similar quality crisis event datasets are available. ACLED has recently expanded its data collection actives in the Middle East and Asia.

Continuous growth in data coupled with new machine learning applications, from domains as diverse as remote sensing and medical sciences, has lead to new challenges and opportunities for predictive analytics. The use of machine learning to develop predictive models requires significant repetitive work to configure algorithms and optimize solutions. Developing effective machine learning systems is a time-consuming
and complex task, even with expert-level knowledge. Manually optimizing predictive accuracy requires selection of the best algorithm with proper tuning of their hyperparameters. Instead of manually building and evaluating a handful of models, an automated system can evaluate hundreds of thousands of models to discover and recommend an optimized pipeline that reaches or surpasses human expert-level performance.

Automated machine learning allows data scientists and researchers to spend their time being creative and gaining a deeper understanding of the problem and the relevance of model outputs, rather than on monotonous tasks that are highly automatable. Research in the automated machine learning space is anticipated to evolve rapidly over the next several years. As these tools mature and become more widely available, it is anticipated that scientific discovery will accelerate with a whole new class of users experimenting with machine learning applications. This will permit researchers to better spend their time analyzing data, exploring assumptions, and evaluating results versus developing and optimizing complex machine learning pipelines.

There is no single learning algorithm that will short-cut the development process and deliver optimal performance for all problems and data. While some methods have excelled at certain application, such as representation or deep learning methods on speech recognition and computer vision problems, they also require an enormous amount of data and compute resources. Other learning methods can often produce better results on a variety of problems, and are not dependent on massive, labeled datasets. A particular algorithm may be good for one data type and size, but not for others. The No Free Lunch
(NFL) theorem states that results that demonstrate that if an algorithm performs well on a certain class of problems, then it necessarily pays for that with degraded performance on the set of all remaining problems (David H. Wolpert, 1996b, 1996a). In other words, any two algorithms are equivalent when their performance is averaged across all possible problems (D. H. Wolpert & Macready, 2005). The effect of the NFL theorem was empirically and automatically evaluated in this study across a variety of datasets. No single method proved to be optimal in all cases.
CONCLUSION

Generally, people use anonymity-granting tools because they are concerned about their privacy and want to avoid governments and companies infringing on their civil rights. This research set out to measure the escalation in Tor usage during periods of civil unrest. To better understand this relationship, a 5-day window on Tor usage was captured in the modeling process. These time lags accounted for the different temporal usage patterns that may exist prior to or during protests. The analysis incorporated nine African countries covering a diverse set of geographic regions, event types, and actor categories. The results demonstrated that a reasonably strong relationship exists between social movements and Tor usage. This was the first study to produce quantitative analytic results that demonstrate that anonymizing tools are essential for protecting human rights and used for important societal purposes.
REFERENCES


event data. *Political Geography, 41*, 1–10. https://doi.org/10.1016/j.polgeo.2014.03.001


BIOGRAPHY

Brian Sandberg graduated from Boardman High School, Boardman, Ohio, in 1984. He received a Bachelor of Science in Applied Mathematics from Kent State University in 1990. He served nine years with the United States Air Force Air National Guard as a Command and Control Specialist. His research interests include geospatial science, machine learning, data science, and anonymity technologies. He founded Conarch LLC in 2009, where he conducted research and development for multiple agencies including Naval Research Laboratory, Office of Naval Research, and National Reconnaissance Office. Brian joined Defense Advanced Research Projects Agency (DARPA) Information Innovation Office (I2O) as scientific, engineering and technical advisor in November 2013.