

A Case Study of Applying Data Mining to Sensor Data for Contextual Requirements Analysis

Angela Rook, Alessia Knauss, Daniela Damian, Alex Thomo

Department of Computer Science

University of Victoria

Victoria, British Columbia, Canada

arook@uvic.ca, alessiak@uvic.ca, danielad@cs.uvic.ca, thomo@cs.uvic.ca

Abstract—Determining the context situations specific to contextual requirements is challenging, particularly for environments that are largely unobservable by system designers (e.g., dangerous system contexts of use and mobile applications). In this paper, we describe the application of data mining techniques in a case study of identifying contextual requirements for a context-aware mobile application to be used by a team of four long-distance rowers. The context of use for this application was dangerous and isolated, making it unobservable by the developers. The context situations for five mobile application requirements were defined by using a data mining algorithm applied to historical sensor data passively collected by the users while they crossed the Atlantic Ocean in a rowboat. The performance of the resulting classifiers is analyzed over time with promising results demonstrating that the data mining approach is feasible with implications for requirements engineering, context-aware mobile applications, and group-context-aware mobile applications.

I. INTRODUCTION

A common problem for mobile application developers is that the context of use of the application cannot always be anticipated at design time [26], and therefore an incomplete set of user requirements is a result. It is challenging to cost-effectively maintain system relevance through manually updating and evolving system requirements. Additionally, even if all system contexts of use can be anticipated at design time, user requirements and the specific *context situations* they are associated with are constantly evolving at runtime [9]. In order to fulfil *context-aware* system requirements, the context situations that the requirements are valid within need to be monitored for by the system at runtime. The automatic definition of these context situations would help with the reduction of system operational and maintenance costs.

The research presented in this paper aims at investigating *how machine learning (data mining in particular) can be applied to sensor data in order to better understand unobservable system operating environments to support the requirements elicitation process*. More specifically, the investigation demonstrates how the context situations relevant to specific *contextual requirements* (i.e., those requirements only valid within specific contexts [14]) can be automatically captured by applying data mining algorithms to historical contextual sensor data. This is shown through a case study of a context-aware mobile system for a group of four long-distance rowers who operate in unobservable and dangerous environments (such as crossing the Atlantic Ocean in a modified rowboat).

The development of this system presented requirements engineering challenges that were addressed by applying offline data mining techniques to historical contextual sensor data passively collected during previous operation of the system.

This study is completed in two steps. First, the J48 (C4.5) data mining machine learning algorithm is applied to historical sensor data to automatically identify which context situations are relevant to five specific requirements. Next, the results of applying the data mining algorithm to historical sensor data is compared against the actual context situations in which specific requirements are valid. The results of this approach inform a discussion of the application of data mining algorithms in this domain, as well as the application of the approach to other domains.

Section II describes the background of this research including some ways that data mining is currently being applied to context-aware systems and some of the challenges faced. Section III contextualizes the particular requirements engineering problem faced by the designers of the context-aware system from the case study. It also describes how the research was carried out including data collection, how data mining was applied to the data, and the results of the process. Section IV discusses how the data mining approach addressed the requirements engineering problem faced in Section III, as well as how the approach may be applied to other domains. It also discusses the limitations of the study. Section V concludes the paper and discusses potential areas for future research.

II. BACKGROUND

The purpose of this section is to define the terms used in this paper and to situate the case studied in this paper in the current state of the art.

A system that is able to adapt itself to changing context is called a *context-aware* system. A system is considered *context-aware* if it can process and apply context information to adapt its functionality to the current context of use [9], [12]. While there may be contextual information relevant to the system above and beyond that which the system can detect (e.g., the “larger social context” or “community” [1]), it is only that context which the system can *detect* and *process* that can be used in practice [9]. Context-aware applications can reduce the amount of effort a user has to put into interacting with an application and can automatically deliver desired services

[13]. For example, if an application can sense the current context situation, then it can efficiently adapt automatically without the user having to take any action. Additionally, sensor technology calibrated to a specific user's profile can enable them to leverage that context for a wide range of services [18]. There is currently great interest in context-aware mobile applications in health-related fields where users are empowered to make health-conscious decisions based upon the activity-related sensor input the context-aware system receives and interprets for them [18], [21], [25], [23]. The UbiFit Garden project is an example of a health-oriented context-aware application that displays continuous, ambient updates about the activity levels of a user so that they know how much physical exercise they have completed during the day [18].

One of the ways that context awareness is supported is by integrating *data mining classifiers* [16], [23]. *Data mining* is a form of machine learning that generally uses historical data to form statistical predictions about future data [17], [27]. That is, data mining classifiers are used to identify real-life *situations* such as “at home”, “running”, or “imminent danger” [10]. Data mining algorithms are often applied when human analysis is not feasible (e.g., very large amounts of data), and also to discern subtle and non-obvious patterns in data. A data mining *classifier* is derived by applying a data mining algorithm (such as the RIPPER [31], [5] or C4.5 [24] algorithms) to a *training set* of representative data. The resulting classifier is then applied to new data in order to classify it. In the case of context awareness, data mining classifier inputs include context sensor readings (singly or aggregated), and outputs include a classification (often represented as a binary value) that the context-aware system can then use to decide whether or not a service should be delivered to the user [3], [6], [8].

Much work has been undertaken to produce classifiers by data mining historical context data for context-aware applications for mobile systems [22], [23]. This includes deriving and performing empirical comparative analyses on classifiers from different movement patterns from sensors (e.g. sleeping, walking, running) in order to keep track of daily activities for health monitoring applications [18], [23], [2]. There are many examples of context-aware systems that implement classifiers to provide services to the user [22]. One example is the mobile-based *easyLife* application which recommends context-relevant businesses to the user based on the user's current time and location [32]. There is also current work in mobile context awareness focusing on monitoring computing context in mobile devices in order to make better use of mobile system resources while data mining streaming context [10].

Users expect context-aware systems to anticipate and respond to their requirements as unobtrusively and correctly as possible. In order to do this, a context-aware system must be able to detect changing situations, and correctly meet requirements associated with the current situation. Unfortunately, it is difficult for designers to anticipate all the context situations that a context-aware system in a dynamic environment will be operating in. This is especially true of ubiquitous and mobile systems where contexts of use may be constantly changing

[29], [18], [15]. As such, context classifiers for context-aware applications on mobile devices need to evolve to continue to reflect the context situations they represent so that user requirements can continue to be met [29], [18], [15], [4], [7], [9], [28], [6]. This means that data mining algorithms should be applied to context training data to update a context classifier when concept drift has occurred to the point that the context classifier is no longer able to adequately determine when it's associated requirement should be fulfilled.

III. THE CASE STUDY

This section reports on the application of data mining algorithms to historical sensor data from the context of use of a group of rowers from OAR Northwest¹. This was for the purpose of discerning the context situations relevant to specific requirements for a context-aware mobile application.

A. The Requirements Engineering Problem

OAR Northwest is a Seattle, USA based non-profit organization who undertake long-distance rowing voyages in order to perform research and deliver science, technology, engineering, and math curriculum to classrooms online and through school visits. The voyages OAR Northwest undertake are completed on a custom rowboat designed for long-distance, open-water journeys, and are propelled entirely by the rowers themselves.

Collaboration with OAR Northwest started in September 2012 in order to develop a context-aware mobile system for an open-ocean voyage from Dakar, Senegal to Miami, Florida, USA. The need for context-awareness within the system developed was justified by the rowers for two reasons. The first of these was because of the extreme and dangerous conditions they often faced on the open ocean. The rowers expressed that, ideally, the system should be aware of these conditions and adapt accordingly in a non-obtrusive way. Additionally, the rowers often faced extreme fatigue and wished to use the system for cognitive offloading such that the system would ‘think’ for them in a number of circumstances so as to better support them in achieving their research and voyage-completion goals. Context-awareness was seen as a way to support cognitive offloading in this manner.

Because of the highly isolated and dangerous nature of the system context of use, it was impossible for software developers to observe the rowers interacting with the system in order to perform typical requirements engineering activities such as ethnography. This created the requirements engineering challenge of how to elicit system requirements, and the context situations that those requirements are active in, for users operating in environments that are unobservable by system developers. This challenge prompted the exploration of how the passively collected sensor data that OAR Northwest collected during their voyages may be investigated for additional insight for the requirements engineering process.

¹<http://oarnorthwest.com/>

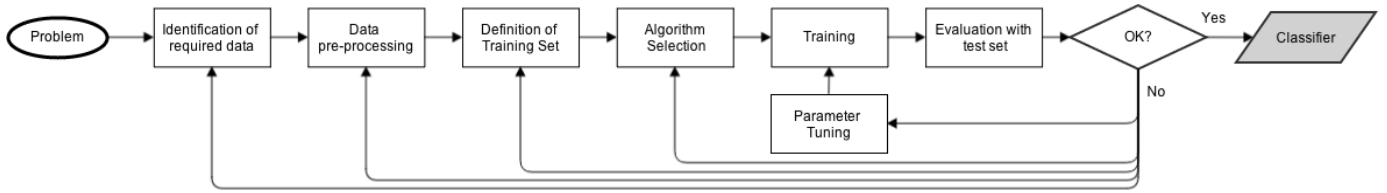


Fig. 1. Kotsiantis' described approach to data mining [17].

TABLE I
CONTEXTUAL REQUIREMENTS INVESTIGATED IN THIS STUDY

Req't	Context Situation Data Mined	Context-Aware Application Action (Contextual Requirement)	How Data Mining Target Attribute was Defined
CR1	<i>on sea anchor resting</i>	<i>automatically disable system alerts</i>	visual inspection of sensor data and verification with log data
CR2	<i>two rowers are sleeping</i>	<i>automatically enable "wake-up" alerts only</i>	function based on <i>Asleep or Awake</i> sensor data for all four rowers
CR3	<i>mentally fatigued but still rowing</i>	<i>automatically assign less cognitively challenging activities</i>	visual inspection of sensor data and verification with log data combined with <i>Mental Fatigue</i> sensor data threshold of 0.2 or less for each rower
CR4	<i>on sea anchor active</i>	<i>automatically assign non-rowing activities</i>	visual inspection of sensor data and verification with log data
CR5	<i>one rower is sleeping</i>	<i>automatically set alerts to visible only (no audio alerts)</i>	function based on <i>Asleep or Awake</i> sensor data for all four rowers

B. Contextual Requirements and Sensor Data

To study *how machine learning can be applied to sensor data in order to better understand unobservable system operating environments to support the requirements elicitation process*, five requirements from a trans-Atlantic row and the context situations specific to them were analyzed. This involved applying a data mining algorithm to passively collected sensor data to produce classifiers which resulted in the definition of the context situations specific to each requirement analyzed.

The case study data set covers a period of 64 days from a trans-Atlantic row attempt from Dakar, Senegal to Miami, Florida, USA that OAR Northwest undertook in early 2013. Data was collected by onboard sensors that recorded biometric and environmental data including passively collected sensor data from a GPS sensor, biometric readings from all four rowers on board the boat from Fatigue Science² Readibands, as well as environmental readings from an Airmar PB200 WeatherStation. The data was delivered after the voyage was completed to the researchers by the rowers and other scientific research organizations partnering with OAR Northwest. Sensor data collection was as consistent as safety and equipment operation allowed, although there were some sensor loss and gain due to technical difficulties.

C. Data Mining Process to Define Situations in which Contextual Requirements are Valid

We used the general data mining approach described by Kotsiantis (see Figure 1) for the specific purpose of identifying context situations relevant to specific requirements (described in Table I). The process consisted of the following (iterative) steps:

1) *Identification of Required Data*: This involved the collection of the reified data that may be relevant to each requirement's active/triggered and inactive/not triggered states. In this study, passively collected sensor data (i.e., rowers' biometric data from Readibands, local environmental information from the Airmar and GPS sensors) was gathered and included in the data mining process.

2) *Data Pre-Processing*: The collected data was then *integrated* (merged into a single table), *cleaned* (erroneous data including outliers were removed, gaps in data due to frequency differences in sensor readings were filled), *transformed* (data was put into numeric formats that could be processed by the data mining algorithm), and *reduced* (columns that were empty, identical, and contained primarily unique data values were removed from the data set according to data mining best practices). This resulted in a data set of 90748 rows with 40 columns of sensor data. Examples of the sensor data included atmospheric temperature, relative wind speed, and each rower's actigraphy (motion sensing) and whether they were asleep or awake.

3) *Definition of Training Set*: Classifiers were independently trained in cumulative increments of approximately every three days of data for a total of 22 independent classifiers produced for each of the five requirements. In the approach taken in this study, the mapping between user requirements and reified context was implemented for data mining through the *target attribute* (also called an *indicator attribute*). Each requirement had a unique target attribute associated with it. This target attribute was represented for data mining as an additional column in the table of sensor data with a binary label for each row indicating whether or not the requirement it represented was active/triggered (1) or inactive/not triggered (0) for that row. In this study, the target attribute was inferred from the data set in one of the following three ways for each

²<http://fatiguescience.com/>

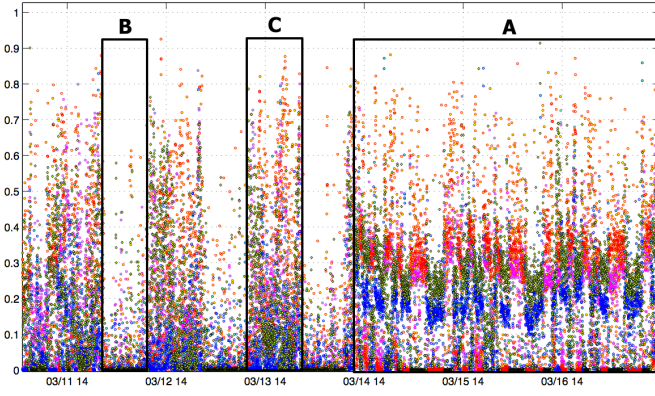


Fig. 2. Graph of combined actigraphy for all four rowers demonstrating examples of (A) normal rowing behaviour in shifts by teams of two rowers at a time, (B) *on sea anchor resting* context, and (C) *on sea anchor active* context.

requirement:

- Through mathematical derivation based on some numerical threshold or function (CR2, CR5),
- Through visual inspection of the sensor data in graphical form (e.g., Fig. 2) and correlating log data with anomalous patterns in the sensor readings (CR1, CR4), and
- Through a combination of the first two (CR3).

Visual inspection of the log data was necessary in order to validate that the data patterns observed were, in fact, indicative of the desired context situations for CR1, CR3, and CR4. The target attribute derived for each requirement was also validated by the rowers during an interview.

4) *Algorithm Selection*: The algorithm selected was chosen based on its comprehensibility to requirements engineers and users [17]. It was also chosen because of its suitability to the data set based on a previous exploratory study. The J48 algorithm is a Weka algorithm based on the well-known C4.5 algorithm [24].

5) *Training*: Context classifiers were produced by training the J48 data mining algorithm on the historical, passively collected, runtime sensor data described above for each of the five requirements (CR1 to CR5) shown in Table I. Training the classifiers for each requirement was accomplished using Weka 3.6.9, a data mining application [11].

6) *Evaluation with Test Set*: All classifiers were evaluated using stratified ten-fold cross validation [30]. This evaluation technique is used to determine the general performance of the classifier by breaking the training set into ten parts (folds), training on the union of nine of those folds using the desired data mining algorithm, and then testing on the one remaining fold. This process is repeated, one for each fold, and the performance results of all ten folds are averaged to obtain a reduced-variance estimate of performance rates on the training set [17]. The performance metrics produced in Weka that are averaged using stratified ten-fold cross validation for each classifier include precision, recall, and f-measure. Daily log data from the voyage and interviews with the rowers were also used to evaluate and validate the results.

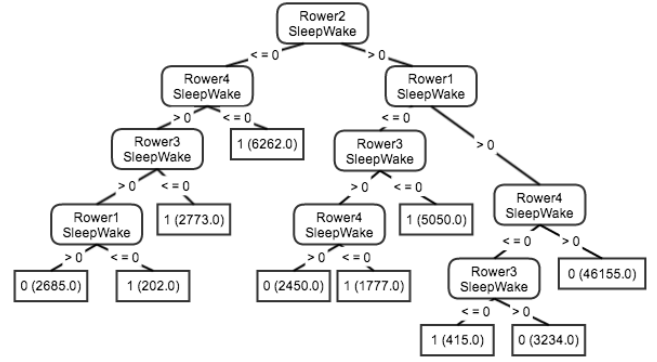


Fig. 3. Visualization of J48 decision tree context classifier produced for CR2 at point 18 from the time-series analysis described in Section III-D.

7) *Classifier*: The context classifier produced at each of the 22 points in the time-series analysis (shown in Fig. 4) define the context situation for each requirement at that point. That is, when specifically a requirement should and should not be active/triggered in the context-aware system is defined by the resulting decision tree classifier based on the historical contextual sensor data. This decision tree can then be analyzed by a requirements engineer to better understand the unobservable context that the requirement should be active/triggered and inactive/not triggered in. An example of such a decision tree is shown in Figure 3. The branch made by the top node and the one to its immediate left can be interpreted as ‘*if Row2 is sleeping and Row4 is sleeping, then requirement CR2 should be triggered/active*’ with a coverage of 6262 rows of data in the training set.

D. Validation of Results

Validation of the resulting classifiers is presented in Figure 4, and was conducted through a time-series analysis that was used to visualize the performance of the classifiers produced by the J48 algorithm as the data in the training set increased over time. These classifiers were independently trained in cumulative increments and then tested on all remaining (future) data in the set for each increment. The graphs show how the J48 algorithm would have predicted the context situations for each contextual requirement had the algorithm been applied to produce new context classifiers every three days plus times of significant sensor configuration changes at runtime. There were 22 classifiers produced, evaluated, and validated for each of the requirements in Table I.

Declines in the performance of the algorithm shown in Figure 4 indicate significant changes in the context that defines the active/triggered and/or inactive/not triggered situations for a given requirement. The descending slope before the local minimums indicates that the previous context data for the requirement is becoming less relevant to the future context of use. The slope of the ascent after a minimum indicates how quickly the algorithm is able to define new, relevant context for the future data. The greater the slope, the more quickly the algorithm can define new, relevant context situations for

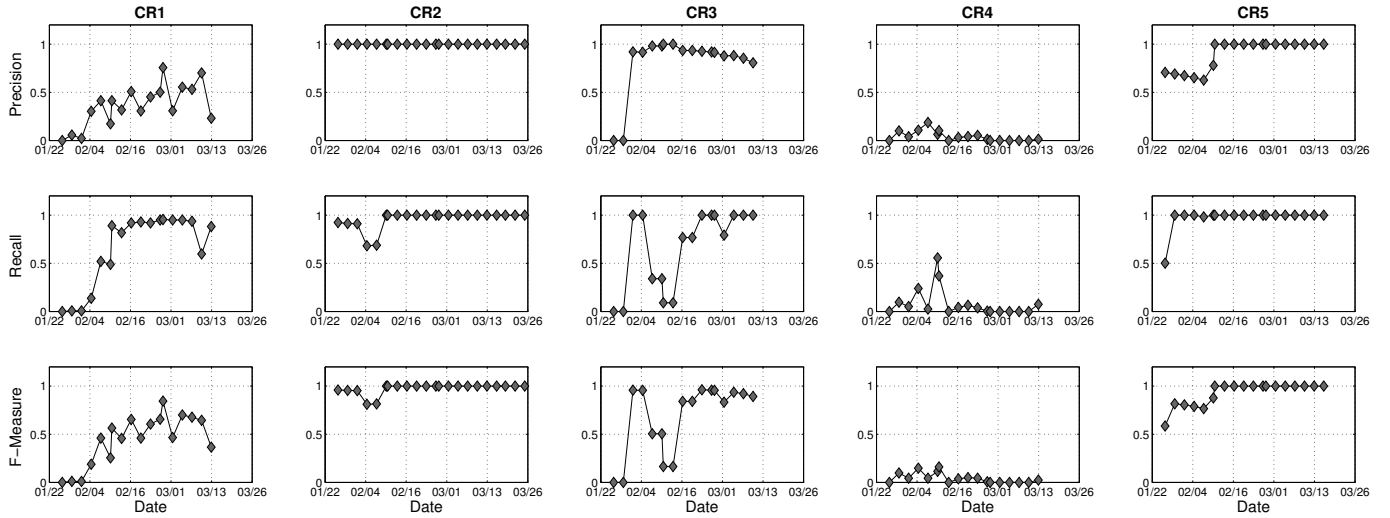


Fig. 4. J48 algorithm performance analysis on runtime data over time for the requirements from Table I. Shown are the precision, recall, and f-measure of correctly identified instances in the test set (all instances in the data set after the date of the given point) from the classifier created on the training set of all instances in the data set before the date of the given point. The closer the point is to the value ‘1’, the better the performance of the classifier produced at that point on all future data (after that point) in the set.

the requirement, and thus provide context awareness for that requirement more effectively.

Surprisingly, sensor configuration changes did not have as much impact on the performance of the context classifiers as expected. However, the introduction of new active/triggered state data for a given requirement had a much greater impact than anticipated. While it was expected that new data such as this would cause an immediate increase in performance, the opposite often happened. However, these dips in performance were often immediately followed by a sharp rise in performance again. This might indicate that a ‘settling time’ after receiving new active/triggered state data may be desirable before generating a new context classifier. Additionally, a sliding window for the reintroduction of old sensor data after a sensor comes back online may be desirable so as to reduce noise, as in the case of the reintroduction of Airmar data for CR1 at point 13.

Although the target attributes for CR1 and CR4 were visually defined in very similar ways, the performance of the precision and recall was significantly lower for CR4. This could be due, at least in part, to the fact that CR1 had approximately twice the active/triggered state data that CR4 did (see Table II). However, it could also be due to the fact that the algorithms simply could not adequately discern differences between the active/triggered and the inactive/not triggered states. Additionally, bias may have been introduced during the visual inspection of the data, and this may have also contributed to the lower performance of CR1, CR3, and CR4. Pattern recognition algorithms, such as unsupervised clustering algorithms, may help improve performance in these cases.

1) *Target Attribute for Context Situations:* In the approach taken in this paper, all the target attributes contain binary values, and the binary classification pertains to whether or not the context aware application should or should not deliver a

TABLE II
NUMBER OF ACTIVE/TRIGGERED STATE, INACTIVE/NOT TRIGGERED STATE, AND UNKNOWN STATE ROWS IN EACH OF THE REQUIREMENTS EXAMINED FOR A TOTAL OF 90748* ROWS IN THE DATA SET.

Req’t	Active/Triggered State	Inactive/Not Triggered State	Unknown State
CR1	8190	82559	0
CR2	21270	66471	3008
CR3	8531	82218	0
CR4	4005	86744	0
CR5	14561	66334	9854

*note that a small boundary error in the script used to divide the test and training sets resulted in an overlap between the sets of one row

particular service for that instance. The data mining algorithms use the target attribute and sensor data to derive a classifier by identifying patterns from the correlations between the target attribute and the training data.

How to effectively correlate requirements with valid context situations through the target attribute was one of the major concerns of this study. A target attribute that is truly reflective of the valid context situations for each requirement is critical to the data mining algorithms’ ability to define context situations from sensor data for those requirements. Intuitively, it is also vital for ensuring that the context derived by the data mining algorithm is truly representative of the context that the requirement is active/inactive in so that system context-awareness is properly implemented.

In order to ensure that the data mining algorithms produced accurate and relevant context from the contextual sensor data provided and the derived target attributes, the target attributes were verified and validated through multiple means, when possible. These included looking for patterns in the sensor data that indicated relevant context for a requirement (see Figure 2), and then verifying that those patterns were representative of

the context for the requirement under investigation (verifying and validating with log data and interviews with the users).

Ideally, the target attribute would be passively captured by the system when the requirement is active/triggered, and inactive/not triggered. Input may also be captured from the users as to when the requirement should be active or inactive. However, neither method on its own may be entirely precise, (thus, introducing noise into the target attribute and producing less accurate results with the data mining algorithms). Therefore, target attributes for requirements for context awareness should be correlated to passively collected sensor data and active user input for verification and validation when possible.

IV. DISCUSSION

The validation of results presented in Section III-D show that the data mining approach presented in Section III-C was useful to varying degrees for better understanding unobservable system operating environments to support the requirements elicitation process. Context situations for the active/triggered and inactive/not triggered states for five requirements were defined using the J48 (C4.5) data mining decision tree algorithm applied to historical contextual sensor data sets. There were several implications for using data mining for Requirements Engineering, for Context-Awareness for Mobile Applications, and for Group-Context-Aware Mobile Applications, which are described in the following sections.

A. Data Mining and Requirements Engineering

There were several influences on the results. Data pre-processing proved to have a significant impact, particularly data reduction. Additionally, the characteristics of the data set including how consistent the data collection was, how much data was available, how representative the training sets were of the test sets, and how consistent the user goals were over the period the sensor data was collected all impacted the results, with more generally producing more accurate results.

The separation between the active/triggered state and the inactive/not triggered state for each requirement (not only in the target attribute, but also in the contextual sensor data), appears to have an impact on the results, particularly in the case of CR4. Additionally, the target attribute for each requirement was not captured by the system at runtime and was instead defined, verified, and validated in one of three different ways, depending on the requirement (see Section III-C). The possible discrepancy between the target attributes derived in this study and target attributes that would have been captured at runtime is unknown. Therefore, the impact that this discrepancy would have on the results is also unknown.

The results of the time-series analysis were used to address how much historical contextual sensor data needed to be collected in order to define the active/triggered and inactive/not triggered situations for each of the requirements. The time-series analysis also addressed how sensor configuration changes and other significant contextual changes impacted the performance of the context classifiers.

The reduction of physical contextual sensor data from the training sets (i.e., the Airmar sensor environmental data at point 7 and the biometric data at point 20) didn't affect the results of the classifiers as much as expected. However, the addition of active/triggered state data was shown to have a loss of performance impact on the classifiers in several cases, which was unexpected. Concept drift may have been notably present in one case (CR1) upon the reintroduction of the Airmar sensor data into the training set at point 13 after a period of approximately two weeks where it was unavailable.

Distribution of the active/triggered state throughout the data, as well as how the target attribute was defined may have an impact on the performance of the classifiers. However, because the sample size of the analysis is limited to within five requirements, it cannot be said for certain.

B. Data Mining and Context-Awareness for Mobile Applications

As discussed in Section II, data mining has been shown to be useful for identifying context situations for service delivery in context-aware applications. The isolated and dangerous nature of the system context of use in the case study presented in this paper made it impossible to perform recommended requirements elicitation activities (e.g., [13]) in order to determine the contexts that the user groups might be situated in. Instead, passively collected runtime sensor data, daily log activities, and interviews with the rowers were relied upon for requirements elicitation. Data mining proved to be a useful means of identifying subtle context of use situations for context-aware service delivery for several requirements.

In addition, location was not considered as relevant to our context-aware application for the requirements we investigated because the locations were unique throughout the duration of the application's use in the contexts investigated (long distance, open-water rows). Nor does this study take into account contextual information provided through wireless means (ubiquity) because the users were in an extremely isolated environment with wireless connectivity being prohibitively expensive (i.e., the Atlantic Ocean). There are many existing studies and context-aware applications centred on leveraging location context for mobile ubiquitous systems (e.g., [3], [15]). This study gives an example of a context-aware application that is not only primarily dependent on non-location sensor data alone, but also gives a real example of an application where the context-aware mobile system cannot leverage wireless connections.

Some of the studies mentioned in Section II also feature actigraphy as a useful physical and user context for context-aware applications (e.g., [18], [23]). This study centred around biometric sensor data (physical and user context) from each of the four rowers, as well as physical contextual data from two environmental sensors. While the environmental sensor data did appear in many of the classifiers produced, far more important in the ranking was the sensor data obtained directly from the users. This is interesting because although the weather may be one of the causes of the context-aware

service being delivered, the physical effect it had on the rowers appeared to be far more relevant to the algorithm's definition of the classifier. This was demonstrated in the case of CR1 where high wind and waves influenced when the rowers were *On Sea Anchor*, but the most important rule in one of the classifiers (i.e., those with the most coverage), had to do with when three of the rowers were described as being *In Bed* by their biometric sensors. This confirms the importance of biometric sensors for individual users, and may be an indicator that they are more important than environmental sensors to deriving context situations for context-aware applications, even when the environment is the cause of the current context situation. It is known that processing data on resource-constrained devices such as mobile phones is challenging [10]. By prioritizing the sensors that are monitored (e.g., choosing to monitor the biometric sensors and potentially dropping the environmental ones entirely), resources on mobile devices can be better optimized.

C. Group-Context-Aware Mobile Applications

The context situations investigated were complex and unique, often depending on the data from more than one user at a time. Many previous studies rely on defining context situations for application service delivery based on movement activities from a single user, rather than a group of users (e.g., [18], [23]). While this study does, indeed take into account biometric information from a single user in the case of CR3, the other four requirements consider context situations for the entire group of four users.

Group context-aware mobile applications are an emerging area of interest with implications for emergency first responders, teams, and military field personnel in order to help them complete missions [19], [20]. Two of the requirements in this study were directly based on hostile environmental conditions (CR1 and CR5), and CR3 was based on the user's physical and mental condition in that hostile environmental condition. The study presented here could have implications for this domain, particularly since the context of use in this study is on the "tactical edge" (i.e., in a resource-limited and hostile environment), as are many of the applications in this emerging field.

D. Limitations

1) *Internal Validity*: Threats to internal validity were mitigated in this study by pre-processing the data set, and then reducing the independent variable in the data set to the indicator attributes for each requirement. It should be noted that CR2 and CR5 had slightly smaller data sets due to the number of rows of data where it was unknown whether they were in the active/triggered state or inactive/not triggered state (see Table II).

Significant effort was made to validate and verify the target attributes for each of the requirements (described in Section III-C); however, it was not possible to determine the target attributes' actual correlation to the context each of the requirements was active/triggered and inactive/not triggered

in. Regardless, the ultimate goal of this study was the investigation of how context-aware systems can automatically adapt to changes in when requirements are typically active/triggered and inactive/not triggered. That is, one of the assumptions is that the target attribute for a given requirement will be subject to change over time, regardless of how it was initially defined. Even so, an attempt was made to minimize the impact of erroneous target attributes on the analyses by completing them on multiple requirements.

2) *External Validity*: The setting of the case study was almost experimental as the rowers operated under normal conditions in very isolated and physically extreme conditions with a small number of variable external impacts. Given this, there may be some question as to how the results may change in a more subtle environment with less clear differentiation between active/triggered and inactive/not triggered states for each requirement. Nonetheless, the runtime data used in this study is from real users interacting with a system in a real context of use.

Additionally, while the users were operating in a restricted environment with similar activities being completed consistently over time, the results may also be applicable to other contexts with consistent patterns of activity. These might include other contexts that rely on shift work (such as hospital wards, or factories), teams of athletes, or even military personnel. Additionally, given the results for CR3, the results may also be applicable to individuals acting with consistent patterns of activity.

V. CONCLUSION AND FUTURE WORK

It is challenging to capture context situations relevant to specific contextual requirements, especially in some mobile and other unobservable contexts of use. In this paper, we presented a case study on a context-aware mobile system for which the context of use was unobservable. Data mining was used on historical sensor data in order to define the measurable context situations in which five particular requirements should be executed by the context-aware system. We validated the resulting classifiers produced through the data mining process in a time-series analysis based on real data from the context of use of a trans-Atlantic rowing trip by the users of the system. The results were promising, demonstrating that the data mining approach is appropriate for defining context situations in which contextual requirements should be executed. Our results could have implications for the context-aware mobile domain, and the group context-aware mobile application domain.

Our results are based on one case study. Further case studies are needed where data mining algorithms are applied and evaluated for similar, isolated situations, as well as to other domains (e.g., healthcare context-aware systems).

ACKNOWLEDGMENT

The authors would like to thank the following people and organizations for their contributions to the research undertaken in this study. The members of OAR Northwest who participated in this study, including Jordan Hanssen, Markus Pukonen,

Adam Kreek, Patrick Fleming, and Greg Spooner. Dr. Fritz Stahr from the University of Washington School of Oceanography and Fatigue Science (<http://www.fatiguescience.com>) for providing the data sets used and their invaluable insight in preprocessing them. Prof. Hausi Müller from the University of Victoria Department of Computer Science for his invaluable insight and expertise in Self-Adaptive Systems. Also Jason Cummer for his work on the development of the context-aware mobile application used in this study.

REFERENCES

- [1] D. Allen, S. Karanasios, and M. Slavova. Working with Activity Theory: Context, Technology, and Information Behavior. *Journal of the American Society for Information Science and Technology*, 62(4):776–788, 2011.
- [2] L. Barnard, J. S. Yi, J. A. Jacko, and A. Sears. An Empirical Comparison of Use-in-Motion Evaluation Scenarios for Mobile Computing Devices. *International Journal of Human-Computer Studies*, 62(4):487–520, 2005.
- [3] J. Burrell and G. K. Gay. E-graffiti: Evaluating Real-World Use of a Context-Aware System. *Interacting with Computers*, 14(4):301–312, 2002.
- [4] B. H. Cheng, R. Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, B. Becker, N. Bencomo, Y. Brun, B. Cukic, G. Marzo Serugendo, S. Dustdar, A. Finkelstein, C. Gacek, K. Geihs, V. Grassi, G. Karsai, H. Kienle, J. Kramer, M. Litoiu, S. Malek, R. Mirandola, H. A. Müller, S. Park, M. Shaw, M. Tichy, M. Tivoli, D. Weyns, and J. Whittle. Software Engineering for Self-Adaptive Systems: A Research Roadmap. In B. Cheng, R. Lemos, H. Giese, P. Inverardi, and J. Magee, editors, *Software Engineering for Self-Adaptive Systems*, volume 5525 of *Lecture Notes in Computer Science*, pages 1–26. Springer Berlin Heidelberg, 2009.
- [5] W. W. Cohen. Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.
- [6] J. Coutaz, J. L. Crowley, S. Dobson, and D. Garlan. Context is Key. *Commun. ACM*, 48(3):49–53, Mar. 2005.
- [7] A. K. Dey, G. D. Abowd, and D. Salber. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Hum.-Comput. Interact.*, 16(2):97–166, Dec. 2001.
- [8] N. Eagle and A. (Sandy) Pentland. Reality Mining: Sensing Complex Social Systems. *Personal Ubiquitous Comput.*, 10(4):255–268, Mar. 2006.
- [9] A. Finkelstein and A. Savigni. A Framework for Requirements Engineering for Context-Aware Services. In *First International Workshop From Software Requirements to Architectures (STRAW 01) 23rd International Conference on Software Engineering*. IEEE Computer Society Press, 2001.
- [10] P. D. Haghighi, S. Krishnaswamy, A. Zaslavsky, M. M. Gaber, A. Sinha, and B. Gillick. Open Mobile Miner: A Toolkit for Building Situation-Aware Data Mining Applications. *Journal of Organizational Computing and Electronic Commerce*, 23(3):224–248, 2013.
- [11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The Weka Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.
- [12] D. Hong, D. Chiu, V. Shen, S. Cheung, and E. Kafeza. Ubiquitous Enterprise Service Adaptations Based on Contextual User Behavior. *Information Systems Frontiers*, 9(4):343–358, 2007.
- [13] D. Hong, D. K. W. Chiu, and V. Y. Shen. Requirements Elicitation for the Design of Context-Aware Applications in a Ubiquitous Environment. In *Proceedings of the 7th International Conference on Electronic Commerce, ICEC '05*, pages 590–596, New York, NY, USA, 2005. ACM.
- [14] P. Inverardi and M. Mori. Requirements Models at Run-time to Support Consistent System Evolutions. In *IEEE Int. Works. on Requirements@Run.Time*, pages 1–8, 2011.
- [15] E. Kaasinen. User Needs for Location-Aware Mobile Services. *Personal Ubiquitous Comput.*, 7(1):70–79, May 2003.
- [16] P. Korpip, M. Koskinen, J. Peltola, S.-M. Mkel, and T. Seppnen. Bayesian Approach to Sensor-Based Context Awareness. *Personal and Ubiquitous Computing*, 7(2):113–124, 2003.
- [17] S. B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press.
- [18] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell. A Survey of Mobile Phone Sensing. *Communications Magazine, IEEE*, 48(9):140–150, Sept 2010.
- [19] G. Lewis, M. Novakouski, and E. Snchez. A Reference Architecture for Group-Context-Aware Mobile Applications. In D. Uhler, K. Mehta, and J. Wong, editors, *Mobile Computing, Applications, and Services*, volume 110 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 44–63. Springer Berlin Heidelberg, 2013.
- [20] G. A. Lewis, S. Simanta, M. Novakouski, G. Cahill, J. Boleng, E. Morris, and J. Root. Architecture Patterns for Mobile Systems in Resource-Constrained Environments. In *Military Communications Conference, MILCOM 2013-2013 IEEE*, pages 680–685. IEEE, 2013.
- [21] C. Liu, Q. Zhu, K. A. Holroyd, and E. K. Seng. Status and Trends of Mobile-Health Applications for iOS Devices: A Developer’s Perspective. *Journal of Systems and Software*, 84(11):2022 – 2033, 2011. Mobile Applications: Status and Trends.
- [22] Q. Liu, H. Ma, E. Chen, and H. Xiong. A Survey of Context-Aware Mobile Recommendations. *International Journal of Information Technology and Decision Making*, 12(01):139–172, 2013.
- [23] J. Parkka, M. Ermes, P. Korpipaa, J. Mantjarvi, J. Peltola, and I. Korhonen. Activity Classification using Realistic Data from Wearable Sensors. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):119–128, Jan 2006.
- [24] J. R. Quinlan. *C4. 5: Programs for Machine Learning*, volume 1. Morgan Kaufmann, 1993.
- [25] C. Samuels. Sleep, Recovery, and Performance: The New Frontier in High-Performance Athletics. *Physical medicine and rehabilitation clinics of North America*, 20(1):149–159, Feb. 2009.
- [26] N. Seyff, F. Graf, P. Grnbacher, and N. Maiden. Mobile Discovery of Requirements for Context-Aware Systems. In B. Paech and C. Rolland, editors, *Requirements Engineering: Foundation for Software Quality*, volume 5025 of *Lecture Notes in Computer Science*, pages 183–197. Springer Berlin Heidelberg, 2008.
- [27] P.-N. Tan, V. Kumar, and M. Steinbach. *Introduction to Data Mining*. Pearson Publishing, 2005.
- [28] N. Villegas, G. Tamura, H. Müller, L. Duchien, and R. Casallas. Dynamico: A Reference Model for Governing Control Objectives and Context Relevance in Self-Adaptive Software Systems. In R. Lemos, H. Giese, H. A. Müller, and M. Shaw, editors, *Software Engineering for Self-Adaptive Systems II*, volume 7475 of *Lecture Notes in Computer Science*, pages 265–293. Springer Berlin Heidelberg, 2013.
- [29] H. Wang, W. Fan, P. S. Yu, and J. Han. Mining Concept-Drifting Data Streams using Ensemble Classifiers. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03*, pages 226–235, New York, NY, USA, 2003. ACM.
- [30] S. M. Weiss and C. A. Kulikowski. *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. M. Kaufmann Publishers, San Mateo, USA, 1991.
- [31] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
- [32] J. Zhuang, T. Mei, S. C. Hoi, Y.-Q. Xu, and S. Li. When Recommendation Meets Mobile: Contextual and Personalized Recommendation On The Go. In *Proceedings of the 13th International Conference on Ubiquitous Computing, UbiComp '11*, pages 153–162, New York, NY, USA, 2011. ACM.