Practical Considerations for Testing the Verdande Technology Use Case

December 6, 2014

1 Verdande Technology: Test and evaluation

1.1 Use-case requirements

The test and evaluation procedures for Verdande Technology will be developed along the lines introduced in Deliverable 2.1: Instead of testing each use case separately, we utilize the notion of *application scenarios*. An application scenario is defined by a sequence of use cases that combined constitutes a full interaction procedure leading to a verifiable result. In D2.1 we defined two scenarios:

- VER1: Detection of drill string vibrations: The application scenario covers these use cases from (D1.2):
  - UC1: Parsing data sets
  - UC2: Erratic torque detection (model construction)
  - UC5: Model application
  - UC6: Result generation
- VER2: Semi-automatic labelling: The application scenario covers these use cases from (D1.2):
  - UC1: Parsing data sets
  - UC3: Semi-automatic labelling (model construction)
  - UC5: Model application
  - UC6: Result generation
- VER3: Automatic formation detection: The application scenario covers these use cases from (D1.2):
  - UC1: Parsing data sets
  - UC4: Formation detection (model construction)
  - UC5: Model application

  - UC6: Result generation

Requirements for the different use-cases were defined in D1.2. Most requirements are functional in nature, and introduce functionality requests into the system, but some also introduce hard requirements that can be tested quantitatively. The latter are repeated in 1 for completeness. We note that the requirements centers around three issues:

- The evaluation process for application scenario VER1 will be defined in this document (requirements VER.U2.D2 and VER.U2.D3).
- In application scenario VER2, the rate of mis abnormalities and rate of missed abnormalities must be better than the priors distributions and should be better than an inexperienced driller (requirements VER.U3.D2, VER.U3.D3, VER.U3.D4 and VER.U2.D5).
- In appliction scenario VER3, mean distance from the estimated formation shifts to the ground truth is essential (requirements VER.U4.D2 and VER.U4.D3).

1.2 Model and data characteristics

In order to make this document self contained, the summary of the Verdande Technology use case scenarios are repeated below.

1. **Automatic detection of drill string vibrations and abnormal torque states** This task aims to better diagnose the shape of the wellbore and the state of the equipment, make better decisions on how to manage the well, and thereby reduce the non-productive time. Currently, the state-of-the-art algorithm in DrillEdge is based on detecting erratic torque. It is a simple algorithm based on comparing root mean squared differences in a time window between the measured torque and a smoothed version of the torque. In essence, this method is measuring the local amplitude variations in the torque. Additionally, the algorithm is altering out abnormal torque during connections and some other instances. This simple algorithm is rather limited and does not inherently take into account the existing uncertainty in the streaming data. The main objective for this scenario is then to design a probabilistic graphical model for erratic torque monitoring and detection of abnormal situations.
2. **Semi-automatic labelling** The main contribution from DrillEdge [28] during operation is that it enables the driller to see in real time if the current situation is *similar* to previous situations that lead to undesired circumstances. If this is the case, remedial actions can be found to avoid undesired events. At the core of the analysis are historical examples of noteworthy episodes, ranging from clearly undesired events, for instance getting stuck, to more subtle indications of evolving problems, such as abnormal readings of hook-load. Verdande Technology has already extensive datasets, where such episodes are identified. However, the de

ID	Sub-phase	Description	Task(s)
VER.U1.O1	Software development	Data import should take less than 10 minutes on a single desktop computer or equivalent	7.1
VER.U2.D2	Production testing	Formalization of the test procedure will be developed in task 1.2.	1,2 / 7.2
VER.U2.D3	Production testing	The accuracy of the result and the discussion should be published in a journal or a conference.	7.2 / 9.3 / 9.4
VER.U3.D2	Production testing	The rate of missed abnormalities must be no larger than the prior probability of abnormality.	7.2
VER.U3.D3	Production testing	The rate of missed normal states must be no larger than the prior probability of normality.	7.2
VER.U3.D4	Production testing	The rate of missed abnormalities could be below what is obtained by an inexperienced drilling engineer or a software program at that level.	7.2
VER.U3.D5	Production testing	The rate of missed normalities could be below what is obtained by an inexperienced drilling engineer or a software program at that level.	7.2
VER.U4.D2	Production testing	MD algorithm must be less than MD prior (defined in Use Case Scenario 3 in Delivery 1.2), on at least 1 randomly chosen drilling log.	7.3
VER.U4.D3	Production testing	MD algorithm should be 10 percent less than MD prior (defined in Use Case Scenario 3 in Delivery 1.2) on at least 3 randomly chosen drilling logs.	7.3
VER.U5.D4	Testing	Agents need to run fast enough for real-time implementation. In practice, this means that once probability tables are calculated (post learning), the agent once provided a single new data point must be able to calculate probabilities, values or classifications and returning these in under 1 second on a single desktop computer or equivalent hardware	3.3 / 7.3

Table 1: Testable requirements for the Verdande use-cases.

nitions of the episodes are not always crisp, and domain experts can at times disagree about what is happening at a certain time. This can potentially lead to inconsistencies in the dataset, complicating both the automatic learning of models as well as the comparison with previous situations. Additionally, it can confuse the driller to be presented with information that he/she does not relate to or agree with. Manually labelling new data sequences, or adapting the existing labelling to new definitions, can be a very time-consuming activity. Verdande Technology is therefore interested in looking into techniques for semi-automatic labelling. This is currently not part of the DrillEdge system in Verdande. Given unlabelled data streams collected over time from typical drilling conditions, semi-automatic labelling aims to compute a normality score for each considered drilling situation, then label it as either *normal* or *abnormal*. As for the previous task, a probabilistic graphical model will be designed, taking into account the temporal dynamics of the drilling process and continuously adapting to changes in the incoming streaming data.

3. **Automatic formation detection** This task aims to predict in real time the formation tops from the MWD (measurements while drilling) data using a proba- bilistic graphical model. Once again, this should be performed taking the temporal dynamics of the drilling process into account. The automatic formation detection is vital for dealing with several issues such as hole instability and vibrations, and also important for reducing the costs and the overall non-productive time. To Verdande Technology's knowledge, this has never been done before, neither as an academic exercise nor as in a product.

1.2.1 The data generation process

The data set that is available from Verdande is complex and involving many variables that will not be used in the chosen use case scenarios. Moreover, mnemonics and units are generally not consistent. In order to simplify these issues, all the drilling logs are simplified to two data sets that will be used directly by the Amidst software. Data set one is used on the use case scenario one and two, while the second data set is used on use case scenario three.

1.2.2 Dataset one

This data set contains 100 drilling logs of various size. Typically one log covers one drilling section or approximately a few weeks of data. For each of the 100 logs two XML files have been generated. The first XML file contains a subset of graphs with common mnemonics and SI units. The XML files contains a fixed number of variables , where the resolution of the logs are below ten seconds between updates.

Among the variables that are contained in the XML file is the normality index, the torque vibration index and also the depth log indicator.

The normality index is identified automatically by DrillEdge. The condition is taken to be normal, when no case is on the radar and no events have fired in a time window of 10 minutes (forward and backwards) and in a depth window of 5 meters (upwards and downwards).

Torque vibration index (no, yes, NA) are basically identified on logs that are fully tagged.

The depth log indicator is pointing out which time stamps are elements in the second XML file. It is basically, the time steps of drilling (except when the depth is manually adjusted). The second XML file is made by simply filter based on the depth log indicator.

The second XML file contains a subset of graphs with common mnemonics and SI units on a depth scale. This subset is using the activity code (that is when we are drilling) to filter out data when non drilling appears. In this case, each data point represent only one depth. The depth logs are unique, meaning that only one measure for each depth. (In the case when the depth is reset manually to a lower number, the measurements between the two candidate depths are deleted.) The XML files contains a fixed number of variables, which are indexed on the depth they were drilled.

1.2.3 Dataset two

At the current state only one data log that is available for use case scenario 3. This is a time log from OMV. This log is manipulated to be run in the AMIDST software. It is expected that more logs will come from OMV and that Verdande will provide sufficient manipulation of the new logs as well. This data log is different than the logs in data set one because it contains down hole measurements such as gamma ray and resistivity. Moreover, this data log also contains lithology charts from planning and after the well is drilled.

For each data log two XML files have been generated. The first XML file contains only a subset of graphs with common mnemonics and SI units on a time scale. The XML files contains a fixed number of variables , where the resolution of the logs are below ten seconds between updates.

The second XML file contains a subset of graphs with common mnemonics and SI units on depth scale. This subset is using the activity code (that is when we are drilling) to produce a set of depth logs as discussed above. The XML files contains a fixed number of variables, which are indexed on the depth they were drilled.

1.3 Predictive performance: test and evaluation

1.3.1 Application scenario 1

The goal of the first application scenario is to detect abnormal torque states. In essence the output of the dynamic Bayesian network is a probability of abnormal torque at each time step. The evaluation involves comparing this output stream to torque vibration index on logs that are fully tagged.

The tags are time intervals that are approximately two to ten minutes long. Even though experts have tagged them, they are not clear cut. For instance, two different experts may disagree on both whether a tag is real or not and also the start and end points of each tag. However, in this project the set of tags are taken to be the ground truth.

Two requirements are relevant for testing this use case scenario.

1. VER.U2.O2 Formalization of the test procedure will be developed in task
2. VER.U2.D3 The accuracy of the result and the discussion should be published in a journal or a conference.

As a consequence the evaluation method must *comply* with the state of the art methods on one side and make sense from a practical point of view on the other side.

It is important to note that detecting torque vibrations early is generally *not* very important. When vibrations are present the driller feels the vibrations easily and is not dependent on being told by a software system. The real value is to automatically detect the vibrations and be able to diagnose how much cumulative damage has the equipment been subject to. We have therefore designed an evaluation method that only takes into account if the tag is detected and the ability to detect an event early is ignored.

The AUC measure is of interest. A set of negative tags, with time windows equal to the time windows of positive tags, are generated from the logs by random. It is important that the negative tags are taken far enough away from the the positive tags so no dependencies are present.

Basically, as the algorithm run through all the logs, the maximum value of the output function is stored for every positive and negative tag. The AUC is computed as of equation (citation). It is an aim to be above 0.90.

1.3.2 Application scenario 2

The goal of the first application scenario is to detect abnormal states. In essence the output of the dynamic Bayesian network is a probability of abnormality at each time step. The evaluation involves comparing this output stream to the normality index on all 100 logs.

The tags are time intervals that are in ranges of hours. These tags are clearly not clear cut as they are based on the output of a complex software system. In this project the set of tags are taken to be the ground truth.

These requirements are relevant for testing this use case scenario.

1. VER.U3.O2 The rate of missed abnormalities must be no larger than the prior probability of abnormality.
2. VER.U3.D3 The rate of missed normal states must be no larger than the prior probability of normality.
3. VER.U3.O4 The rate of missed abnormalities could be below what is obtained by an inexperienced drilling engineer or a software program at that level.
4. VER.U3.D5 The rate of missed normalities could be below what is obtained by an inexperienced drilling engineer or a software program at that level.

In order to meet these requirements it is important to define true positives, false positives, true negative and false negatives. A set of negative tags, with time windows equal to the time windows of positive tags, are generated from the logs by random. It is important that the negative tags are taken far enough away from the the positive tags so no dependencies are present.

Basically, as the algorithm run through all the 100 logs, the maximum value of the output function is calculated for every positive and negative tag. The classification rule is to compare these output values (maximum on each tag) to a given threshold.

In order to compare with an inexperienced drilling engineer, an evaluation will only be done on three logs. The drilling engineer is instructed to tag regions which he believes are abnormal.

The above measures are global and can be estimated after the logs have been processed. The performance measures do not take into account concept drift.

In case of drilling a section (the data contained in one log), there is clearly concept drift. To be more precise, a drilling operation starts out normal, but as the open hole is getting longer, the probability of abnormality increases as the hole gets more unstable and also more dirty. Moreover, equipment are expected to be in good shape as the operation starts, but degrades with drilling.

It is therefore of interest to compute the prequential AUC with a forgetting factor (as of equation xx) and plot this versus time. A proper discussion of this is believed to be of interest.

1.3.3 Application scenario 3

In this application scenario it is of interest to both find the formation tops and also to correctly label them. From Delivery 1.2 we have these requirements:

1. VER.U4.D2 MDalgorithm must be less than MDprior (both defined in Use Case Scenario 3), on at least 1 randomly chosen drilling log.

2. VER.U4.D3 MDalgorithm should be 10 percent less than MDprior on at least 3 randomly chosen drilling logs.

The measure MDprior is the mean deviation between the expert identified formation tops before drilling and after drilling. The measure MDalgorithm is the mean deviation between the formation tops that are detected by the algorithm and the formation tops that are identified by the experts after the well is drilled. Clearly, it is a goal to make an algorithm where MDalgorithm is less than MDprior.

If we can assume that all formation shifts that are present in the litology chart from planning, actually happen, even though not exactly the same places, then it should be possible to force the algorithm to have the same formations as the ground truth. In this case, finding an algorithm to calculate the mean deviation should be straigh forward.

However, this task is not this easy. In particular, one can not assume that the sequence of formations in the planning lithology is exactly the same as the sequence in the ground truth. We therefore need to make a measure that allows for a penalty of adding or removing an extra formation.

In general, I don ot have the answer for this. Helge suggested looking at sequence alignment on wiki. I tried, but did not figure it out. Will think more. A simple option would be to assume that sequences from planning are the same as the ground truth, by some manipulation of the data.

1.4 Runtime performance: test and evaluation

The requirements below are related to runtime performance. These requirement are fairly straightforward to accomplish. We will therefore only list them to give a holistic exposition.

1. VER.U1.O1 Data import should take less than 10 minutes on a single desktop computer or equivalent.

2. VER.U5.D4 Agents need to run fast enough for real-time implementation. In practice, this means that once probability tables are calculated (post learning), the agent once provided a single new data point must be able to calculate probabilities, values or classifications and returning these in under 1 second on a single desktop computer or equivalent hardware