“THEME ARTICLE”, “FEATURE ARTICLE”, or “COLUMN” goes here: The theme topic or column/department name goes after the colon.

A Data-analytics Approach for Enterprise Resilience

Donna Xu
Centre for Artificial Intelligence, University of Technology Sydney

Ivor W. Tsang
Centre for Artificial Intelligence, University of Technology Sydney

Eng K. Chew
Design Architecture and Building, University of Technology Sydney

Cosimo Siclari
Information Technology, Reserve Bank of Australia

Varun Kaul
Information Technology, Reserve Bank of Australia

Enterprise resilience plays an important role to prevent business services from disruptions caused by human-induced disasters such as failed change implementations and software bugs. Traditional expert-centric approach has difficulty to maintain continued critical business functions because the disasters can often only be handled after their occurrence. This paper introduces a data-analytics approach, which leverages system monitoring data for the enterprise resilience. With the power of data mining and machine learning techniques, we build an intelligent business analytics system to detect the potential disruptions proactively, and to assist the operational team for enterprise resilience enhancement. We demonstrate the effectiveness of our approach on a real enterprise system monitoring dataset in simulation.

KEYWORDS: predictive data-analytics, data mining, machine learning, enterprise resilience

Organizations (firms and public institutions) seek to attain enterprise resilience so that they can have “capabilities to detect, contain, and bounce back from those inevitable errors that are part of an indeterminate world”[1] to deliver continued business services as required. Enterprise resilience is defined as “the ability of a system to absorb, adapt and recover rapidly from a disruption so that normal levels of the service delivery can resume.”[2] It has a number of properties such as resistance to disruptions, recovery (immediate response to return to stable state) and adaptive learning (environmental fit to a new ‘better’ state).[3] We especially focus on the unexpected disruptions that are caused by human-induced disasters such as failed change implementations, software bugs and etc.

Enterprise resilience plays a vital role in the organizations in providing critical business services, and many resilience analysis frameworks are proposed.[4, 5, 6, 7] Resilient organizations are able to

respond effectively to disruptions and positively evolve to keep pace with the environmental changes. We live in a rapidly changing world. As the enterprise system environment becomes more complex, new disasters and risks will rise, which might result in service disruptions and it will further lead to business function failure. For example, a payment system provided by a financial institution gets upgraded regularly, and the upgrades might unexpectedly result in user account maintenance failure, such as duplicate or missing transactions. If this incident is not detected proactively, it may potentially affect a large number of customers for a period of time. Therefore, resilience is crucial to the continued sustainable performance and stability of the services provided for the organizations.

Most organizations adopt expert-centric approach under the resilience framework.[8] It requires operational team to leverage their expertise and practical experience to conduct a series of resilience processes before, during and after disruptions, such as threats preparation, incident root cause identification, recovery and etc. However, expert-centric approach is ineffective to deliver enterprise resilience to the organization, because being a reactive process it struggles to prevent disruptions. In other words, disruptions are usually reported as incidents by the end-users *after* their happening, and then the operational team start resolving the incidents by invoking the required resilience processes. Therefore, contemporary organizations are seeking new, innovative ways to *proactively* detect impending incidents in real-time to prevent service disruption and to effectively support the operational team to increase their productivity so as to deliver superior enterprise resilience. This is a significant and challenging practical problem with unknown solutions that motivates our research that seeks to design and evaluate a viable solution for enterprise application.

System monitoring data (for server, network, file systems and etc.) plays as a key role for the operational experts to handle the incidents. Usually it is processed based on a set of rules and protocols to provide human understanding information. For example, 90% CPU on a device will be processed and generate the corresponding alarm as critical CPU, to point the experts to the direction for root cause identification. However, such generated alarms result in a loss of information, such as the actual value of a performance metric that is used to generate the alarms, and other performance metrics that imply bad system behavior without the alarm generation. Therefore, the system monitoring data contains a rich source of useful information that could be used to enable real-time *proactive* detection and resolution of impending incidents (before they materialize) and enhance the enterprise resilience. Yet, there is scarcity of knowledge about such kind of data-analytics approaches for enterprise resilience. Hence, in 2017, an Australian financial services organization (coded name FSO) approached UTS researchers to cooperatively investigate (under a three-year contract research) enterprise resilience within their organization; and, in particular, to explore innovative ways by which a predictive analytics could be designed and implemented for FSO enterprise resilience.

Figure 1 illustrates an example of disruption management with a number of procedures of the traditional expert-centric approach compared with our proposed data-analytics approach. The left hand side of the figure shows the steps of the expert-centric approach. It is user reporting driven. We can observe that there is a long period of time between the system showing abnormality (i.e. CPU increases incessantly beyond the safe threshold) and the user encountering the function failure. But expert-centric approach cannot perform any detection during this period. Using the data-analytics approach that is shown on the right hand side of Figure 1, the model in our analytics system learned using system monitoring could detect system abnormality. Warning is then sent to the operational team so that a potential disruption of the service can be prevented. In addition, our analytics system may even provide possible abnormal performance metrics to help the operational team to resolve the issue underlying the potential disruption more effectively. Note that in reality, the abnormal system behavior could be very complex, i.e. not simply a rapidly increasing performance metric.

In this paper, we present a data analytics approach for enterprise resilience. It differs from the expert-centric approach in that it focuses on the use of system monitoring data to build an intelligent business analytics system to facilitate *proactive* detection of abnormal system behavior detection before disruptions. In the proposed analytics system, we define the severity of incidents and formulate the proactive analytics problem to an ordinal classification problem in machine learning. In addition, since incidents in the dataset are unusual events, extreme class imbalance

needs to be handled in our proposed analytics system. Moreover, incidents are reported based on users so it is common to have missing incidents in the incident report and thus in the labeled dataset. Therefore, it is also challenging to deal with noisy ordinal labels in the proposed analytics system. We demonstrate how our proposed system helps the (FSO) operational team to increase their productivity and thus enhance the enterprise resilience. The evaluation results show the effectiveness of our method in incident detection. To the best of our knowledge, we are the first to leverage machine learning and data mining techniques to build a business analytics system for enterprise resilience.

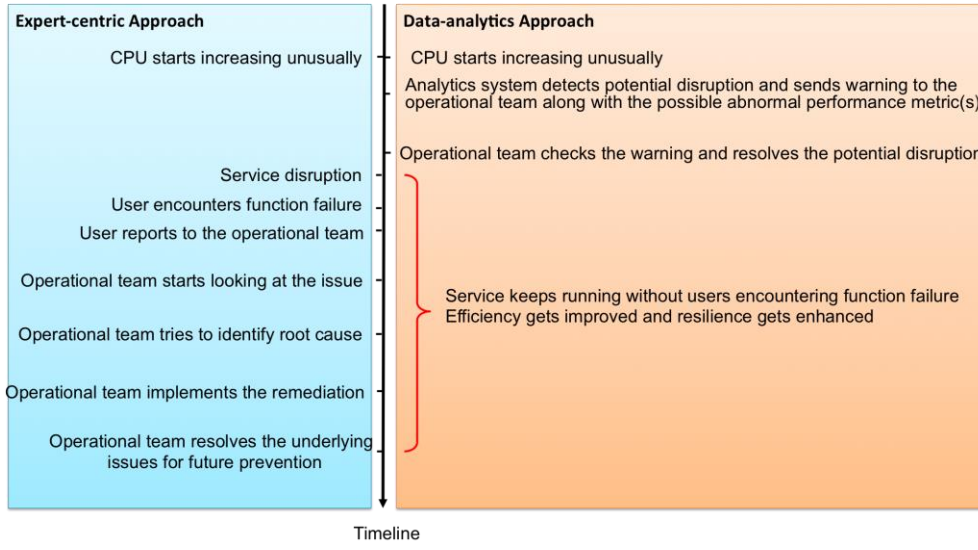

Figure 1. Comparison between traditional expert-centric approach and our proposed data-analytics approach for an example of disruption management.

The paper is structured as follows. Section 2 introduces our data-analytics approach with a number of stages and how these stages map to the processes in the enterprise resilience framework. Section 3 presents our intelligent business analytics system in the perspective of data science. Section 4 evaluates our system. We conclude our paper and suggest several future works at the end.

DATA-ANALYTICS APPROACH IN RESILIENT ORGANIZATIONS

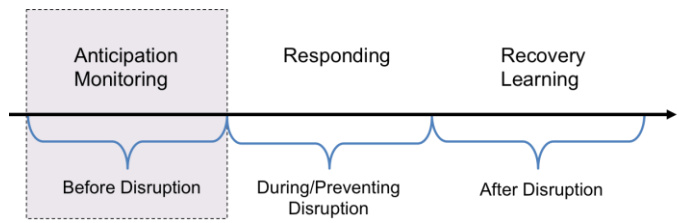

Figure 2. Resilient organization with five processes, anticipation, monitoring, responding, recovery and learning across three phases, before, during/preventing and after disruption.

In resilience engineering and disaster response, the capabilities of enterprise resilience are supported by a number of processes, such as anticipation, monitoring, responding, recovery and learning.[9] The anticipation process requires the domain knowledge to take in the idea about a situation that might potentially occur, in order to cope with the events based on the detection. The

monitoring process acts as surveillance to monitor system parameters for potential environmental disruptions. In the traditional expert-centric approach, the generated system monitoring data are usually rule-based to generate predefined alarms. Such generated alarms could be noisy and difficult to be analyzed manually by experts, so they might limit their power and utilities to help the experts for event discovery.[10, 11, 12] The responding process performs damage control to stop the disruption spreading and causing wider enterprise impacts. The recovery process tries to restore the services from negative events and bring them back to business. The learning process helps to improve the enterprise resilience by learning from the experienced events. Figure 2 presents these five resilience processes before, during/preventing and after disruption. The anticipation and monitoring processes are conducted before disruption. The responding process is conducted during disruption if an unwanted event is detected, or during the prevention of a disruption if an event with potential negative effect is detected. Our proposed analytics system focuses on the phase before disruption.

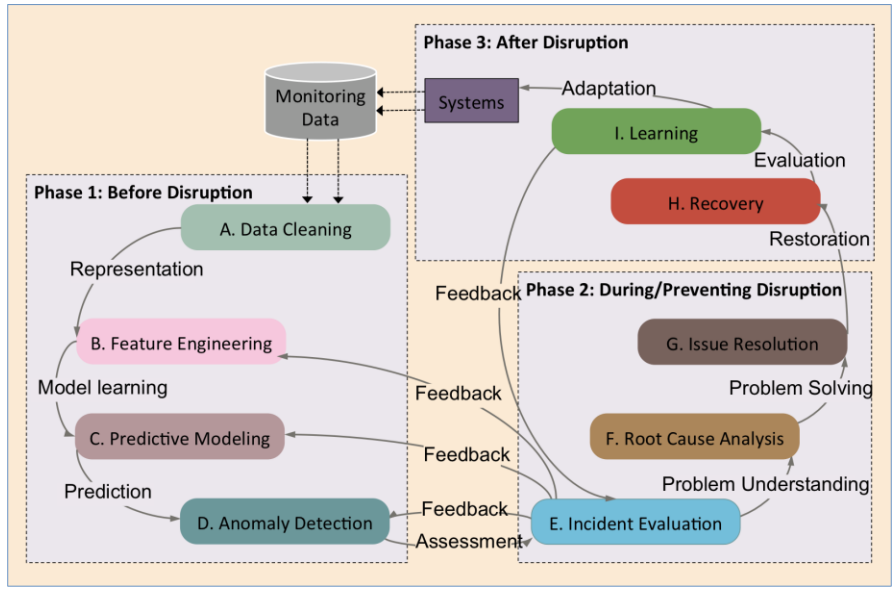

Figure 3. The life cycle of the proposed data-analytics approach for enterprise resilience.

Figure 3 depicts various stages of the life cycle of our proposed data-analytics approach for enterprise resilience on the three phases. The enterprise systems supporting different business services keep generating monitoring data, which is to be analyzed in our approach to help prevent potential disasters and eventually enhance enterprise resilience. We briefly introduce each stage and why it is important to be in the life cycle for enterprise resilience. We focus on the stages A to E.

A. Data Cleaning. System generated monitoring data could be noisy, inconsistent and with missing values. Low quality monitoring data will affect the performance of the predictive model. We manually filter out the data with poor quality (such as incorrect, inconsistent or irrelevant data), and prepare the cleaned data for representation in the stage of feature engineering.

B. Feature Engineering. The quality of the features affect the performance of the predictive model. Feature engineering is the process of the transformation from the cleaned raw data to features that represent the problem to the predictive models, using the domain knowledge of the data. For example, for the predictive task of server incident detection, features can be designed as the performance metrics of the server, such as CPU, memory and so on. Normalization of the feature vector is usually required, to ensure that each feature contributes equally to the model training. In addition, different features usually carry different importance weights to the predictive models and the underlying problems, so it is important to select a subset of features from the original designed feature set, in order to build a more

robust and fast learning model. The transformed and/or selected features can then be used for model learning in the stage of predictive modeling.

C. **Predictive Modeling.** Predictive modeling is the process of using machine learning or data mining for training, testing and validating a model to best make the prediction to the underlying problem. The model is trained based on an algorithm using training data, which is usually all the historical data. Then the trained model can be saved for reused purpose to be applied on the future data for predictions. Multiple models can be trained on different sets of features for different underlying problems. For example, a model can be trained on the features designed as system performance metrics, to detect an incident in the system. Another model can be trained on the features designed for different monitoring data such as system logs, to mine the patterns for system logging information.

D. **Anomaly Detection.** The learned models from stage C with discovered patterns can be used to perform prediction on the future system data to detect abnormal system behavior. However, the abnormal system behavior does not necessarily lead to a service disruption (which makes it to be a false alarm). If an anomaly is detected, a warning will be sent to the operational team for assessment in stage E.

E. **Incident Evaluation.** Given the warning sent from stage D, along with some assistance information such as the selected features, or the discovered patterns on different monitoring data sets, the operational team will evaluate the anomaly, to either start handling the issue if it will potentially lead to an incident or ignore the warning if it is a false alarm. The evaluation results can further be used as the feedback to improve the feature selection in Stage B, update the models in Stage C, or add constraints in Stage D for fewer false alarms.

Stages F-I. The assistance information predicted by different models can be used to help the operational team in the **Root Cause Analysis**, in the way it tells the team what has happened and what aspects of the system possibly went wrong. Then the operational team **Resolve Issues** using their domain knowledge, followed by **Recover** and **Learning** to learn the experience from the anomaly caused by the underlying issue of the system. In addition, the results from the learning stage can further be used as feedback to help the operational team for future incident evaluation in stage E.

INTELLIGENT BUSINESS ANALYTICS SYSTEM

We have shown the data-analytics approach on the operation level. Under the protocol of our proposed data-analytics approach, we present an intelligent business analytics system for incident detection on the implementation level for Phase 1 in Figure 3, to show how machine learning and data mining techniques can be used to help the operational team to enhance the enterprise resilience in the phase of before disruption.

Motivation. To help the experts in handling enterprise resilience, we aim at detecting incidents automatically instead of reporting them by users. A natural solution is to formulate our task into a binary classification problem. Given the historical data, we first transform it into labeled feature vectors, and then use them as input to learn a binary classifier. However, this approach is not suitable for us to perform proactive analytics, because incidents are usually detected before they happen. Proactive analytics requires us to detect potential incidents before they happen. Therefore, it is crucial to define the level of incident severity, especially to define what level for potential incidents. After that, our proactive analytics task can be formulated into an ordinal classification problem in machine learning. Different from independent outputs in multi-class classification, the outputs in ordinal classification have relative ordering. In our case, the outputs represent the severities of the incidents. There are many ordinal classification models proposed to handle the aforementioned problem.[13, 14] However, it is not trivial to adapt them in our setting due to two reasons. First, since incidents are unusual events, there is a high ratio of number of timestamps with non-incidents over the ones with incidents, where extreme class imbalance is required to be handled in the ordinal classification model. Second, the incidents recorded in the incident report are fully based on user reporting. If an abnormal system status happens without any user encounters, there will be an unnoticed incident which will remain unreported. This will lead to noisy labels after the annotation step, where data samples labeled as healthy system status

might actually belong to the issue status. Therefore, to perform proactive analytics, we transform the task into an ordinal classification problem with levels of incident severity predefined. In the proposed analytics system, we adapt the Support Vector Machine (SVM) classifiers[15] to tackle the challenges of class imbalance and noisy ordinal labels.

Our proposed analytics system consists of two main modules. First, a data preprocessing module that translates system monitoring data to ordinal labeled feature vectors. Second, a machine learning module learns the predictive model (an ordinal classifier) and applies the model for potential incidents detection, and selects features to help the operational team identify root cause more effectively.

Module 1: Data Preprocessing

The data preprocessing module aims at transform the raw data into labeled feature vectors. System behavior can be implied in different monitoring data such as server, network and etc. that are generated from different monitoring tools. Apart from the system monitoring data, an incident report is also provided, to indicate the details of each incident, such as time, description, root cause and etc. A change log contains the information about any changes ever made to the service, such as software upgrades. Some incidents happen because of the changes made to the service. The data preprocessing module includes a number of procedures such as data cleaning, feature extraction, normalization and data labeling. The first three steps are explained in detail in the previous section and they are used to obtain the feature vectors. In this section, we demonstrate the data labeling step specifically tailor made in our proactive analytics task.

Data Labeling

To train the machine learning model, we also require the label information for each feature vector at a point of timestamp to show if an incident is happening at the time. Incident report provides the information about the details of each incident, such as the report time of the user, and the issue close time of the operational team. We can label each vector of a timestamp as “issue” if an incident is happening at this time. In addition, we assume that there must be some unusual system behavior right before an incident happens. Such unusual system behavior might be similar to the one during an incident, but it is not severe enough to cause one. Under this assumption, we define the concern label “attention” to label t timestamps before the starting time of each incident. For the rest of the timestamps, we label them as “healthy”. In this way, we have three different output labels, “healthy”, “attention” and “issue”, and they have relative ordering to represent the severity of an incident.

Module 2: Machine Learning

The machine learning module consists of three functions, feature selection, model training and incident detection. Feature selection selects a subset of features that is the most useful or the most relevant to the problem. The selected features can give the operational team a general idea on which ones of the performance metrics have the most influence for the incident detection problem, so as to assist them to find the root cause of an incident and enhance the enterprise resilience. Model training trains the model given the preprocessed input data. In our case, the potential incident detection problem can be formulated as a classification problem with ordinary classes as output. We propose a biased ordinal SVM to solve the proactive analytics task, as well as handle the issues of class imbalance and noisy ordinal labels.

Proposed Biased Ordinal SVM

Support Vector Machine (SVM)[15] is one of the best models for classification problem in terms of classification accuracy.[16] It has many advantages such as scaling well to high-dimensional data and having less overfitting risks. It can be applied to similar applications to ours such as intru-

sion detection[17] or anomaly detection[18]. To perform ordinal classification, we leverage two binary SVM classifiers. The first one classifies data samples with labels between "healthy" and "non-healthy". Apart from "healthy" samples, we use the second classifier to classify samples between "attention" and "issue". To handle noisy labels and class imbalance issues, biased SVM[19] is adopted.

Let $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_n, y_n)\}$ be the set of training examples, where $\boldsymbol{x}_i$ is an input vector and y_i is the corresponding ordinal label. $y_i \in \{0, 1, 2\}$, where 0 represents "healthy", 1 represents "attention" and 2 represents "issue". Let $||.||_1$ $||.||_2$ represent the l_1 and the l_2 norm, respectively. We denote the transpose of vector by the superscript ^{T}. $\boldsymbol{w}$ is the weight vector.

In the first classifier, due to class imbalance, there are only a small number of samples belonging to "non-healthy". In addition, some of the samples with the "healthy" label might belong to the "non-healthy" label due to the unreported incidents. Thus, we are trying to minimize the number of "healthy" examples to be misclassified as "non-healthy" and constrain the "non-healthy" samples to be correctly classified as "non-healthy". In this case, we allow errors for "healthy" samples but do not allow them for "non-healthy" samples due to the noisy labels in "healthy" samples and considerably small number of samples in the "non-healthy" class. The objective of the first classifier is shown in the following SVM formulation.

$$\min \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|_2^2 + C\sum_{i=k}^{n}\xi_i$$

$$s.t.\ \boldsymbol{w}^T\boldsymbol{x}_i + b \geq 1, i = 1,2,\dots,k-1$$

$$-1(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, i = k, k+1, \dots, n$$

$$\xi_i \geq 0, i = k, k+1, \dots, n$$

where C is a positive constant that controls the trade-off between the loss function and the regularizer, and the first $k-1$ samples are with the positive label ("non-healthy").

In the second classifier, which classifies samples between "attention" and "issue" classes, there still exists the issue of class imbalance (from the observation of the given data). In addition, there might be smaller difference in system behavior of these two labels comparing to the one of "healthy" and "non-healthy", so we allow error for samples on both classes. The objective of the second classifier is shown in the following soft-margin SVM formulation.

$$\min \frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|_2^2 + C_+\sum_{i=1}^{k-1}\xi_i + C_-\sum_{i=k}^{n}\xi_i$$

$$s.t.\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, i = 1,2,\dots,n$$

$$\xi_i \geq 0, i = 1,2,\dots,n$$

where C_+ and C_- control the trade-off between the regularizer and the loss function of the samples with the positive label ("issue") and the one with the negative label ("attention").

Feature Selection

Lasso regularization[20] can be used for sparse weighted vector. If we use the SVM formulation of the first classifier, we simply replace the first term $\left|\left|\boldsymbol{w}\right|\right|_2^2$ to be $\left|\left|\boldsymbol{w}\right|\right|_1^2$, to enforce a number of the estimated coefficients in $\boldsymbol{w}$ to be zero, in order to achieve the feature selection. The index of the non-zero coefficients corresponds to the index of the selected features.

Prediction (Incident Detection)

Given a testing sample $\boldsymbol{x}$, we first apply the first classifier on $\boldsymbol{x}$ to predict whether it belongs to "healthy" or "non-healthy" label. If it is predicted as "non-healthy", it will further input to the second classifier to predict whether it is with label "attention" or "issue".

Example of Information Parsing

This section illustrates how information is parsed from raw data to the detection results and feedback through different stages using an example. For a specific service (e.g. payment service), given the raw system monitoring data, we first identify the devices running for this service, and gather all the performance metrics on all the identified devices. Each performance metric on each identified device is designed as a feature, such as CPU, memory and so on. We perform feature selection to select a subset of relevant features so as to remove redundant ones. An example of removed features could be memory utilization on a backup device, as the memory utilization on this backup device is relatively stable with considerably small variance and it has minimum effect to the service disruption. Given the incident report, we perform data annotation to obtain labeled feature vectors. The machine learning module consists of three stages, feature selection, model training and incident detection. Feature selection selects a subset of relevant features. Model training learns the predictive models and the goal of some models may discover patterns such as CPU and memory increase at the same time with similar rate. Incident detection uses the learned model to detect potential disruptions. The selected features, discovered patterns and predicted potential disruptions as the warning will go to the expert for evaluation. The selected features and discovered patterns would help the expert to find the root cause of an incident as it provides the information about what performance metrics on what device are relevant to the incidents. The discovered patterns can also help the expert in improving their logging tools such as system logs or database logs. The warning of detected potential disruption allows the expert to identify the root cause and solve the potential issue before the service gets disrupted. The expert can gradually provide feedback to these three stages for improvement, such as designing new important features, providing more data sources for learning the predictive model, providing feedback on false alarms of the prediction to update the predictive model.

EVALUATION

Our design is based on DevOps, to conduct iterative design and seek feedback from the organizational team. In this section, we demonstrate the evaluation results using a real enterprise system monitoring data set.

Data

We evaluate our model on the system monitoring data provided by FSO. The designed features need to be able to represent the characteristic of the system behavior, such as CPU and memory. In our work, we focus the task of incident prediction, so we design features to be the performance metrics of the systems. Each feature vector represents the performance of the system at a point of timestamp. There are a total of 11,000 data points indexed in time order, with different consecutive time changes (10 minutes, 1 hour or 1 day). There are a total of 21 devices and each device has the monitoring data on 28 performance metrics, on the aspects such as server and networking. Thus, there are 588 (21 x 28) features designed in total for each data point, with each performance metric on a device as a feature.

Simulation Plan

To simulate the predictive model in real-time incident detection, we split the data into training, validation and testing sets. Training set is used to learn the predictive model, validation set is used to tune the parameters of our model, and we simulate testing set as future real-time data to be applied to the learned predictive model for incident prediction.

During the time of these 11,000 data points, there are only 18 data points labeled as “issue”, 23 labeled as “attention” (t is set to be 3) and the rest of them are labeled as “healthy”. To split training, validation and testing set with consecutive timestamps in each of them, we use 90% of the whole data set for training, 5% of it for validation and 5% of it for testing, with 13, 2, 3 “issue” data points, respectively.

Evaluation Metrics

We adopt several commonly used performance metrics in classification problem, precision, recall, F1 and AUC to evaluate our system. We are focusing on evaluating whether true incidents are captured. Recall addresses the question of how well our predictive model can detect the incidents, and precision addresses the question of how likely the predicted incidents by our model are correct. Recall is defined as $\frac{TP}{TP+FN}$ and precision is defined as $\frac{TP}{TP+FP}$,[21] where *TP* represents the number of true positives (i.e. the number of data points labeled as "abnormal" correctly predicted as "abnormal"), *FN* is the number of false negatives (i.e. the number of data points labeled as "abnormal" incorrectly predicted as "normal"), and *FP* is the number of false positives (i.e. the number of data points labeled as "normal" incorrectly predicted as "abnormal").

Recall evaluates whether the true incidents can be detected, so higher values of recall represent better model for incident detection. Precision, on the other hand, evaluates the correctness of the predicted incidents, so lower precision values represent more false alarms. Since it is an ordinal classification problem, we report AUC on each of the classes to represent how well the method distinguishes between each class and the rest of the classes.

Evaluation Results and Discussion

In the testing set, there are 3 data points labeled as "issue" (by incident report). Our predictive model predicts a total of 19 data points as "issue", including the 3 true "issue" data points. Our model achieves 100% recall on "issue" label, which represents that our predictive model is able to detect all the incidents, but with 15.8% precision, which implies that we have many false alarms compared with the incident report.

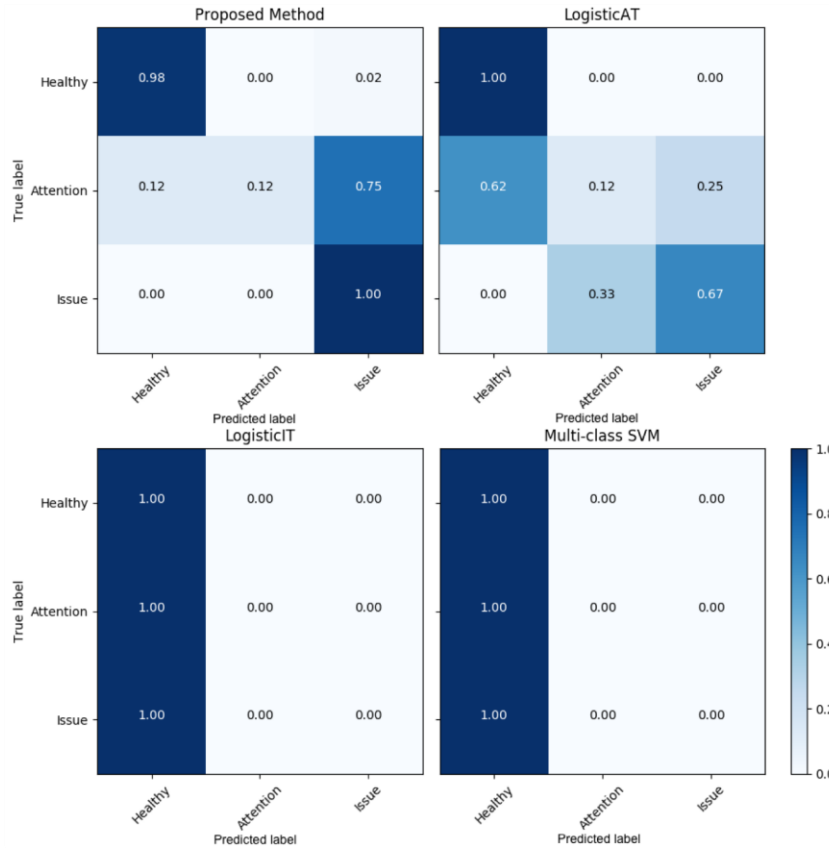

Figure 4. Normalized confusion matrix of our proposed method comparing with three baseline methods in ordinal classification.

We compare our proposed method with three baseline methods, LogisticAT[13], LogisticIT[13] and Multi-class SVM[22]. LogisticAT and LogisticIT are two threshold-based ordinal logistic models. Multi-class SVM treats three output labels as three independent classes. There are many anomaly detection works which adopt deep neural network[23,24] and they can be adapted to solve the ordinal classification problem. However, since they do not consider the concept of "attention" label and deep learning models usually require reasonable amount of data in practice, we do not compare with these methods. Figure 4 demonstrates the results of confusion matrix of all the methods. The values in the confusion matrix are normalized and the values on the diagonal represent the recall value for each label. From the figure, we can observe that our proposed model successfully detected all the incidents with recall on "issue" as 100%. Specifically, it can be seen that LogisticIT and Multi-class SVM classify all the testing data to the dominant class "healthy", which shows that they cannot handle well the class imbalance issue. In addition, LogisticAT incorrectly classifies one of the "issue" samples as "attention", which further verifies that our proposed method can deal with noisy ordinal labels and class imbalance by forcing the hard margin on the "non-healthy" label and allowing different trade-off parameters on the "attention" and "issue" labels respectively. Furthermore, many of the "attention" samples are classified as "healthy" in LogisticAT but they are classified as "issue" in our proposed method. In the following, we present partial results of the predicted incidents compared with the reported incidents, to further evaluate the effectiveness of our proposed method on potential incident detection.

Table 1. The results of Precision, Recall, F1, different classes of AUC for all the methods. The best ones are in bold.

Method	Precision	Recall	F1	AUC		
				Health	Attention	Issue
Proposed Method	**0.7186**	**0.7016**	**0.7100**	**0.9443**	**0.5625**	**0.9840**
LogisticAT	0.6633	0.5972	0.6285	0.7727	0.5615	0.8313
LogisticIT	0.3260	0.3333	0.3296	0.5	0.5	0.5
Multi-class SVM	0.3260	0.3333	0.3296	0.5	0.5	0.5

Table 1 shows different performance metrics for our method and baseline approaches in respect of the FSO data set. The reported metrics include Macro Precision, Recall and F1, and AUC of each of the classes. It can be clearly seen that our method outperforms all the baselines in terms of all the performance metrics. Specifically, the value of the Attention class of AUC indicates that not all the system behavior with *Attention* tag would lead to an *Issue*.

Table 2. Partial results of the predicted incidents compared with the reported incidents (0 represents "healthy", 1 represents "attention" and 2 represents "issue". "Issue" timestamps are highlighted in red color).

Timestamps	Labels by Report	Prediction	Evaluation by
27/3/17 1:54	0	0	N/A
27/3/17 2:56	0	0	N/A
27/3/17 3:57	0	0	N/A
27/3/17 4:56	0	0	N/A
27/3/17 5:58	1	0	N/A
27/3/17 6:59	1	1	□
27/3/17 7:58	1	2	□
27/3/17 8:58	2	2	□
27/3/17 9:56	1	2	□
27/3/17 10:57	1	2	□

27/3/17 11:57	1	2	□
27/3/17 12:59	2	2	□
27/3/17 13:50	1	2	□
27/3/17 14:52	1	2	□
27/3/17 15:52	2	2	□

Table 2 demonstrates the partial results of the predicted incidents compared with the true incidents that are labeled by incident report. From the first and the second columns, we can observe that three incidents (reported by users) happened on the same day. From the third column, it shows that our predictive model has detected consecutive incidents starting from 7:58 to 15:52. The experts then evaluate our predicted results by checking the alarms information on 27 Mar 2017 using their domain knowledge and conclude that the detected "issue" data points belong to the same incident. But these timestamps are not recorded on the incident report, because the incident can only be detected by the report of the users in traditional expert-centric approach. We show the evaluation results by experts in the fourth column of the table, with a tick representing the same incident. It is important to note that, our proposed system is capable of detecting incidents, no matter the types or kinds of the incidents. All the incidents in the incident report are distinct. Some of the incidents may be related in some ways, such as with the same root cause but appear to be different in terms of the incident happened. The detected incident in the testing set is different from all the incidents in the training set. Our results further verify the effectiveness of our proposed method compared with LogisticAT, in which many of the "attention" samples are classified as "healthy", and also verify the effectiveness of our proposed data-analytics approach compared with the expert-centric approach. The expert-centric approach has the limitation in that it cannot record all the incidents and their corresponding time periods, and thus the labels by expert in the second column of Table 2 misses out some true "issue" timestamps. Incidents that are not triggered or encountered by the users (i.e., not recorded in the report) can also be detected using data-analytics approach. Although there are unnoticed and unreported incidents affecting the labelling of the dataset and re-labeling of the dataset would impact the detection capability of the system, we propose the method to handle noisy ordinal labels in our analytics system. It is well verified that our proposed method is able to handle noisy ordinal labels compared with other ordinal classification baselines. In addition, our model detects consecutive incidents around an hour before the first incident-encountered user. Moreover, we have successfully detected an "attention" at 6:59 before an incident to indicate a sign for incident forecasting. Therefore, our proposed data-analytics approach demonstrates the effectiveness and is able to send warning to the operational team earlier than the traditional expert-centric approach, and our proposed system achieves 100% recall for incident detection.

We can further detect whether our predicted consecutive issue timestamps belong to the same incident by the thresholding technique. We compute the distance between each consecutive detected abnormal timestamps, if they are apart by a certain threshold, then we detect them as the same incident, otherwise they belong to different incidents. The verdict of the FSO operation experts verifies our prediction of incidents.

CONCLUSION AND FUTURE WORK

This paper introduced a data-analytics approach for enterprise resilience and presented an intelligent business analytics system. We conducted evaluations to demonstrate that our proposed data-analytics approach is more effective in detecting incidents than the traditional expert-centric approach. Therefore, our proposed method can be used to assist the FSO operational team to improve the enterprise resilience. Our predictive model outperforms baseline methods in terms of precision, recall, F1 and AUC for incident detection in a real enterprise system data set, which further verifies the effectiveness of our method.

There are a number of future works that can be extended under our proposed data-analytics protocol for enterprise resilience. Ouedraogo *et al.* suggest that feedback can be leveraged to improve the enterprise resilience.[25] Therefore, we can incorporate the feedback mechanism in our

system in the future work. From Figure 3, we can clearly see that there are several important problems that can be further studied to improve the whole resilience life cycle. For example, the feedback obtained from Stage E can be used to update the predictive model. This would require the paradigm of online machine learning to get the predictive model updated in real-time. In addition, the feedback can also be used to improve Stage B, to dynamically handle feature selection, so that we could provide the selected features based on the latest data to the operational team to help the team for root cause analysis. Furthermore, time-series features can be studied, to consider temporal information for the predictive model to proactively detect incidents. In the current work we have a limited amount of real-world resilience data with few reported incidents from FSO team, which results in some limitations such that we cannot conduct significance testing or compare with deep learning models in the experiment. We will be able to do so in the future, when we gather more data with reasonable amount of incidents.

REFERENCES

1. K. E. Weick, and K. M. Sutcliffe, "Managing the unexpected: Sustained performance in a complex world," *John Wiley & Sons*, 2015.
2. L. Shen, and L. Tang, "A resilience assessment framework for critical infrastructure systems," *Reliability Systems Engineering (ICRSE)*, pp. 1-5, 2015.
3. E.A.M. Limnios, T. Mazzarol, A. Ghadouani, and S.G. Schilizzi, "The resilience architecture framework: four organizational archetypes," *European Management Journal*, vol. 32, no. 1, pp. 104-116, 2014.
4. R. Francis, and B. Bekera, "A metric and frameworks for resilience analysis of engineered and infrastructure systems," Reliability Engineering & System Safety, vol. 121, pp. 90-103, 2014.
5. R. Filippini, and A. Silva, "A modeling framework for the resilience analysis of networked systems-of-systems based on functional dependencies," *Reliability Engineering & System Safety*, vol. 125, pp. 82-91, 2014.
6. P. Riccardo, G. Di Gravio, F. Costantino, A. Falegnami, and F. Bilotta. "An Analytic Framework to Assess Organizational Resilience." Safety and health at work 9, no. 3 (2018): 265-276.
7. I. A. Herrera, A. Pasquini, M. Ragosta, and A. Vennesland. "The SCALES framework for identifying and extracting resilience related indicators: preliminary findings of a go-around case study." SIDs 2014-Proceedings of the SESAR Innovation Days. EUROCONTROL (2014).
8. S. Huber, I. van Wijgerden, A. de Witt, and S. W. Dekker, "Learning from organizational incidents: Resilience engineering for high-risk process environments," Process Safety Progress, vol. 28, no. 1, pp. 90-95, 2009.
9. J. Lundberg, and B.J. Johansson, "Systemic resilience model," Reliability Engineering and System Safety, vol. 141, pp. 22-32, 2015.
10. S. Jajodia, P. Liu, V. Swarup, C. Wang. "Cyber situational awareness." Springer US, 2009.
11. R. Werlinger, K. Hawkey, K. Muldner, P. Jaferian, and K. Beznosov. "The challenges of using an intrusion detection system: is it worth the effort?." In Proceedings of the 4th symposium on Usable privacy and security, pp. 107-118. ACM, 2008.
12. J. R. Goodall, W. G. Lutters, and A. Komlodi. "Developing expertise for network intrusion detection." Information Technology & People 22, no. 2 (2009): 92-108.
13. J. D. M. Rennie and N. Srebro, "Loss Functions for Preference Levels : Regression with Discrete Ordered Labels," in Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling, 2005.
14. P. A. Gutierrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, and C. Hervas-Martinez. "Ordinal regression methods: survey and experimental study." IEEE Transactions on Knowledge and Data Engineering 28, no. 1 (2016): 127-146.
15. M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines." IEEE Intelligent Systems and their applications, vol. 13, no. 4, 1998, pp. 18-28.

16. M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we need hundreds of classifiers to solve real world classification problems?,” The Journal of Machine Learning Research, vol .15, no. 1, pp. 3133-3181, 2014.
17. W. H. Chen, S. H. Hsu, and H. P. Shen, “Application of SVM and ANN for intrusion detection,” Computers & Operations Research, vol. 32, no. 10, pp. 2617-2634, 2005.
18. T. Shon, Y. Kim, C. Lee, and J. Moon, “A machine learning framework for network anomaly detection using SVM and GA,” Information Assurance Workshop, Proceedings from the Sixth Annual IEEE SMC, 2005.
19. B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. “Building text classifiers using positive and unlabeled examples.” In Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, pp. 179-186. IEEE, 2003.
20. R. Tibshirani, “Regression shrinkage and selection via the lasso,” Journal of the Royal Statistical Society, pp. 267-288, 1996.
21. D. M. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” Journal of Machine Learning Technologies, vol. 2, no. 1, pp. 37–63, 2011.
22. C.-W. Hsu, and C.-J. Lin. “A comparison of methods for multiclass support vector machines.” IEEE transactions on Neural Networks 13, no. 2 (2002): 415-425.
23. C. Zhou, and R. C. Paffenroth. “Anomaly detection with robust deep autoencoders.” Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2017.
24. X. Teng, M. Yan, A. M. Ertugrul, and Y. R. Lin. “Deep into Hypersphere: Robust and Unsupervised Anomaly Discovery in Dynamic Networks.” IJCAI. 2018.
25. K. A. Ouedraogo, S. Enjalbert, and F. Vanderhaegen, “How to learn from the resilience of Human–Machine Systems?,” Engineering Applications of Artificial Intelligence, vol. 26, no. 1, pp. 24-34, 2013.

ABOUT THE AUTHORS

Donna Xu is a PhD student at the Centre for Artificial Intelligence, FEIT, University of Technology Sydney. Her current research interests include multiclass classification, online hashing and information retrieval. Contact her at donna.xu@student.uts.edu.au.

Ivor W. Tsang is an ARC Future Fellow and Professor at University of Technology Sydney (UTS). He is also the Research Director of the UTS Priority Research Centre for Artificial Intelligence (CAI). He received his PhD degree in computer science from the Hong Kong University of Science and Technology in 2007. His research focuses on transfer learning, feature selection, crowd intelligence, big data analytics for data with extremely high dimensions in features, samples and labels, and their applications to computer vision and pattern recognition. Contact him at ivor.tsang@uts.edu.au.

Eng K. Chew is a professor of business IT at the School of Built Environment, University of Technology Sydney. He is a former Chief Information Officer of SingTel Optus and has over 25 years experience in high-tech industries in Australia. His research is industry-focused and centred on the interplay of strategy, innovation and leadership practices in (digital) service innovation. He led/leads industry-based contract research projects on entrepreneurial business innovation and on analytics-based enterprise resilience in Europe and Australia, respectively. He holds B.E. (University of Melbourne) and Ph.D. (University of Sydney). Contact him at eng.chew@uts.edu.au.

Cosimo Siclari is a manager of service management in the Information Technology Department at Reserve Bank of Australia. Contact him at SiclariC@rba.gov.au.

Varun Kaul is a senior analyst of service management in the Information Technology Department at Reserve Bank of Australia. Contact him at KaulV@rba.gov.au.