International Journal of Advanced Science and Technology
Vol. 29, No. 1, (2020), pp. 1109 - 1125

Unsupervised Learning Methods for Anomaly Detection and Log Quality Improvement Using Process Event Log

M. Vijayakamal[1], D. Vasumathi[2]
[1]Research Scholar, Dept. of CSE, JNTUH, Hyderabad, India.
[2]Professor, Dept. of CSE, JNTUHCEH, JNTUH, Hyderabad, India.
[1]kamalmvcse@gmail.com, [2]rochan44@gmail.com

Abstract

With Service Oriented Architecture (SOA) prevailing, enterprise applications of the real world became very complex in size and operations. This has led to several thousands of processes involved. Manual inspection of processes became very tedious and impractical exercise. Process mining came into existence to deal with extraction or discovery of business processes, finding outliers and anomalies besides process enhancement. Out of these aspects, anomaly detection and process enhancement are given high importance. The existing methods in this area have limitations in terms of accurately identifying anomalies and enhancing the processes. To overcome this problem, a comprehensive framework is proposed for detecting anomalies from business process event logs and rectify anomalies for process enhancement. The Simple Auto Encoder based Anomaly Detection (SAE-AD) algorithm is proposed to achieve this. However, it is deterministic in nature while making prediction decisions. A probabilistic algorithm called Probabilistic Auto Encoder based Anomaly Detection (PAE-AD) is proposed to detect anomalies and enhance processes involved. Both algorithms are based on Artificial Neural Networks (ANNs) with unsupervised learning. Empirical study is made with Business Process Intelligence (BPI) challenge datasets especially targeted for process mining research. The empirical results revealed that the proposed framework improves performance in anomaly detection and process enhancement.

Keywords:*–Business process event log, process mining, Business Process Intelligence (BPI), anomaly detection*

1. Introduction

Enterprises record their IT-driven business processes in the form of event logs for different purposes like process monitoring, process enhancement and auditing. Process mining is aimed at extracting knowledge from such event logs. It can discover process models and help in finding outliers or anomalies. The discovered models can be represented graphically using Petrinets or BPMN models or combination of both. Statistical models derived from process models can help in prediction and finding anomalies [1]. Thus predictive monitoring of business process event logs is made possible with process mining. As event logs are complex in nature, machine learning techniques cannot be employed directly to process event logs. In order to use machine learning techniques, the event log data needs to be mapped to a vector space after extracting features [5].

There is exponential growth of data and it needs intelligent means of dealing to ascertain useful knowledge embodied in it. The data recorded in the form of event

ISSN: 2005-4238 IJAST

logs is particularly rich in terms of its utility and the traces left to help uncover anomalies of business processes. Based on the quality of event logs, the analysis yields useful insights that are associated with processed involved in a system. The process of extracting knowledge from event logs of a business process is thus known as process mining [3]. Given some typical examples of executions of processes, it is possible to learn from event logs and have a model to analyse the newly arrived processes. Modelling may take many forms like BPMN or Petrinets [5].

Discovering process models from event logs is an intuitive and impressive mechanism of process mining. However, executable process model is not mandatory as prerequisite. To answer many questions, predictive monitoring business processes can help. Sometimes, it is essential to monitor processes and also find anomalies. By identifying anomalies, they can be rectified by reconstruction of log events. This is one of the much desirable research areas on process mining [4]. From the literature, it is found that mining business process logs is given significance. Different anomaly detection algorithms are explored in [1], [4], [9] and [19]. The notion of contextual anomalies is studied in [2] with multi-perspective anomaly detection. Anomalies that spread across multiple instance is considered in [10].

Anomaly detection in temporal domain is the focus of [5] and [14]. In the existing research different algorithms are proposed to deal with anomaly detection. However, neural network based autoencoder development is found to be efficient. Nevertheless, there is little research on autoencoders development for textual data processing or particularly process event log mining. Therefore, we focused on developing two autoencoders for efficient process discovery, detecting anomalies and enhancing processes. Our contributions in this paper are as follows.

1. We proposed an anomaly detection framework for discovering business processes from event longs, mine anomalies and enhance quality of processes.
2. Two algorithms are proposed. They are known as Simple Auto Encoder based Anomaly Detection (SAE-AD) and Probabilistic Auto Encoder based Anomaly Detection (PAE-AD).
3. An application is built to evaluate the proposed framework. Benchmark datasets from Business Process Intelligence (BPI) challenge are used for empirical study. The results revealed that the proposed algorithms are capable of enhancing processes after discovering business processes from event logs and mining anomalies.

The remainder of the paper is structured as follows. Section 2 provides literature on anomaly detection over process log events. Section provides preliminaries to gain understanding on important terms. Section 3 presents the proposed process mining framework for anomaly detection. Section 4 presents experimental results. Section 5 concludes the paper and gives possible scope for future work.

2. Related Work

Anomaly detection from business process event logs is found to be an important research area with high impact on enterprise businesses. Bezerra and Wainer [1] reviewed many process aware systems with different algorithms for anomaly detection. The algorithms are based on sampling of traces, iterative trace classifier

ISSN: 2005-4238 IJAST

and threshold based approaches. They found that sampling approach is time efficient. Bohmer and Rinderle-Ma [2] focused on contextual anomalies with multi-perspective anomaly detection. They intended to improve their likelihood based method which causes overhead now. Nolle et al. [3] also investigated on multi-perspective based detection. They employed a set of heuristics to determine anomalies. Bohmer and Rinderle-Ma [4] identified different challenges in anomaly detection. They include early detection of anomalies, root cause analysis, looking beyond point anomalies, interoperability, handling changes, unbalanced techniques and evaluation quality.

Rogge-Solti and Kasneci [5] focused on detecting anomalies based on the execution time of events. They used Bayesian model for inferring anomalies automatically. They also detected outliers and measurement errors using ProM library. Conforti et al. [6] proposed an algorithm named FilterLog for finding infrequent behaviours in business process log events to make it noise free. They intended to improve it further with event dependence analysis to find missing logs. Kristof Bohmer[7] used Bayesian belief networks for developing a Decision Support System (DSS) by discovering processes from event logs.

Sureka [8] employed kernel based KNN algorithm for detecting anomalies using unsupervised learning approach. They found the effect of kernel usage with the machine learning algorithm. Bezerra et al. [9] studied Process Aware Information Systems (PAIS) for monitoring business processes in order to fine anomalies. They intended to use genetic algorithms in future for improving the detection process. Bohmer and Rinderle-Ma [10] investigated multi-instance anomaly detection using temporal behaviours of historic events. In future, they intend to focus on more fine-grained details of logs for anomaly detection. Vartiainen et al. [11] studied methods to discovery processes from event logs and find anomalies. Lin et al. [12] proposed a modelling for discovering anomalies from process logs. In their patented work, they generated bipartite graph to have behavioural matrix and use temporal approach to detect anomalies. Wittenstein et al. [13] focused on anomalous recurrent behaviours based on event logs. Their work is patented and it is based on rehistogram concept to find recurring behaviours. Conforti et al. [14] proposed an approach for automatically detecting timestamp errors in event logs and correcting them. It was a probabilistic approach for detecting anomalous timestamps.

Pauwels and Calders [15] used dynamic Bayesian networks for automatically discovering anomalies from business process logs. It is an unsupervised approach that learns from data directly. LeMond et al. [16] proposed a method for finding anomalous accounts from event logs. There are some auto-encoders found in the literature for discovering anomalies in business process event logs. In [19], an autoencoder is used for anomaly detection in Surface Mounted Device (SMD) machine sound dataset. In [20] a variant of autoencoder is employed to detect anomalies in the KPIs associated with web applications. Lopez Martin [21] proposed variational autoencoder to detect anomalies in NSL-KDD dataset as part of intrusion detection. There was little research found on the application of autoencoders for detecting anomalies from business process event logs. In this paper we focused on defining two autoencoders for efficient discovery of business process from event logs, detection of anomalies and then improving processes by correcting anomalies found.

ISSN: 2005-4238 IJAST

3. Preliminaries

Prior to proposing a methodology to discover anomalies from business processes, this section provides prerequisites and definitions to help understand the proposed methodology. **Process** is a collection of activities or events pertaining to a business. When such events are processes in a given sequence it leads to rendering a service. A process is recorded as a case or trace case in the event log. Each process is made up of a set of related events. An **event** is an activity which is part of a business process. **Event log** is a set of cases or traces. Each case corresponds to a particular process involved in the system. **Process mining** is the process of analysing and extracting useful knowledge from business process event log. It is an umbrella term that reflects many techniques to discover knowledge from process event logs. Here are more formal definitions for event log, case and event. Log analysis provides the details of actual processes and their dynamics of implementation. **Applications of process mining** include process discovery, process conformance checking and process enhancement.

Definition 1 (Event log).

Event log is a collection of cases or traces. In other words, it is an n-dimensional vector with a set of cases denoted as L $\in C^n$. Ordering of cases is done for mathematical convenience but it is of no interest.

Definition 2 (Case or trace).

Case or trace is a collection of related events. A set of cases can be denoted as C while CA denotes attribute names of each case. Therefore, an attribute a ∈ CA, case c ∈ C and an attribute value of c is represented as $\#_a$(c). An attribute of a case with missing value is denoted as $\#_a(c) = \perp$. An important attribute of case c is assumed. It is known as trace. This attribute of c is denoted as cˆ =$\#_{trace}(c)$. A trace with a sequence of events is denoted as $\#_{trace}(c) \in \varepsilon *$.

Definition 3 (Event)

An event is an activity of a process or trace or case that is required along with other process to fulfil the purpose of the process. An event is denoted by e and e ∈ε where all possible events is denoted by ε. Every event a has certain attributes denoted as EA and a ∈ EA. The value of a particular attribute of an event e is denoted by $\#_a$(e). If any attribute has a missing value, it is denoted by $\#_a(e) = \perp$. Every event e in event log L has specific attributes. There is an activity attribute denoted by $\#_{act.}$(e) ∈ A where A represents a set of activities. The timestamp attribute is denoted by $\#_{time}$(e) ∈ T where T represents the time domain with temporal ordering. There are some optional attributes such as transaction type, resource associated to an event e and cost associated with e. Transaction type attribute is denoted as $\#_{transe}$(e), resource as $\#_{resourse}$(e) and cost as $\#_{cost}$(e).

The definitions provided here help in understanding process mining easier. Sometimes, it is possible to associate data such as content of a document to attributes of events. Thus in process mining changes to such content can be tracked while processes are being executed. However, in this paper, our focus is only on activities and timestamps. There are no constraints on trace completeness as well and an event log may have ongoing cases.

ISSN: 2005-4238 IJAST

4. Proposed Anomaly Detection Model

We proposed an anomaly detection model based unsupervised neural networks with deterministic and probabilistic models. The aim of the model is to detect process anomalies by mining event log and enhance process by reconstructing log. Figure 1 shows the complete detection model which has two important phases. They are known as anomaly detection phase and event log reconstruction phase. The methodology for anomaly detection takes business process event logs as input. Then it is subjected to pre-processing. The pre-processing generates input matrices as per the underlying algorithms (Section 4.3 and Section 4.4). The input matrices are in compressed format and it is the result of neural network based supervised learning. The input traces are provided to model training which in turn gains capability of anomaly detection. The anomaly detection model generates two anomaly detectors. They are based on event anomaly and time anomaly respectively. Hence it is known as multi-variate detection of anomalies. The detectors can detect anomalies in the log and the result of the anomaly detection phase is the event log with anomalies removed.

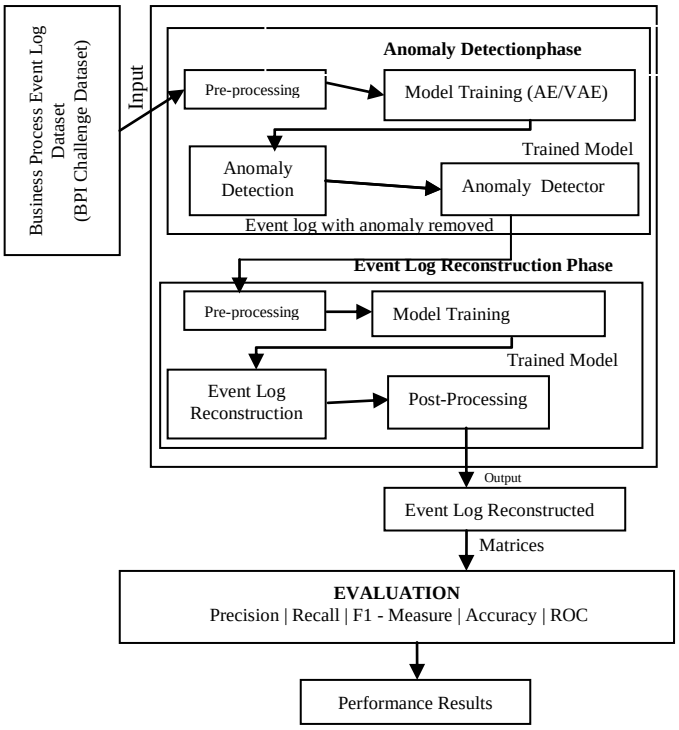

Figure 1. Proposed methodology for detecting anomalies and improving quality of log

Once event log is subjected to removal of anomalies, then event log reconstruction phase begins. Here the pre-processing generates input traces and model training results in a trained model. The training model is used to reconstruct event logs. Event log reconstruction results in output traces that are subjected to post processing in order to see that event log is reconstructed finally. The following two sub sections provide more details of the two phases.

4.1 Anomaly Detection Phase

In this phase, the proposed algorithms (Section 4.3 and Section 4.4)take event log dataset as input. Then a supervised neural network is used to learn compressing of

ISSN: 2005-4238 IJAST

input data efficiently and encode it in a reduced data representation format. The encoding process results in input matrices that are in the compressed format. These input matrices are used to train a model as per the proposed algorithms. Then the trained model is used by the anomaly detection model to produce output matrices. From the output matrices two anomaly detectors are produced. They are known as activity based anomaly detector and time based anomaly detector. Both detectors are employed to have better identification of anomalies and remove them from the event log. This process is illustrated in Figure 2.

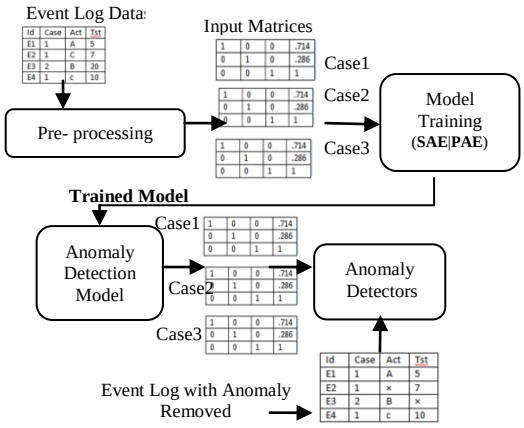

Figure 2. Illustrates the anomaly detection process

As presented in Figure 2, the encoding and decoding procedures produced input and output matrices respectively. The input matrices helped in training a model while the output matrices led to the creation of detectors. As there are two detectors, it is known as multivariate anomaly detection model. Just detection of anomalies will be of less use. It is essential to improve business processes by reconstruction of log. This will have higher utility in the real world applications. The event log reconstruction phase is described in Section 4.2.

4.2 Event Log Reconstruction

The result of anomaly detection phase as shown in Figure 2 is removal of anomalies. The event log with detected anomalies is given as input to this phase of the proposed system. the input data is subjected to encoding process to have a compressed data format in the form of input matrices. The compressed data is given to train a model based on the proposed algorithms (Section 2.3 and Section 2.4). Once the model is trained with the reduced form of data, it can be used to reconstruct output matrices that lead to identification of cases that need post processing to enhance process log. The post processing results in the reconstructed event log.

ISSN: 2005-4238 IJAST

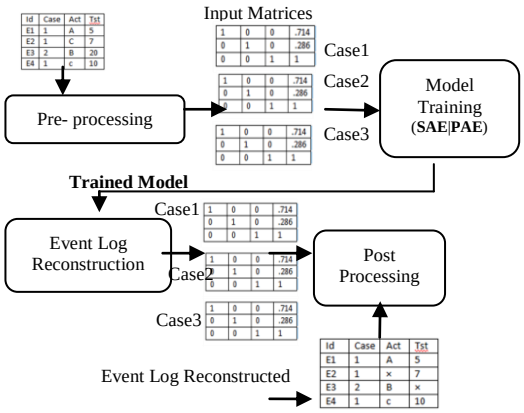

Figure 3. Illustrates the reconstruction of log to improve quality

Reconstructed event log will have no anomalies. Thus the output log is the log with improve quality. Neural networks with deterministic and probabilistic encoding and decoding approaches are realized with the proposed algorithms. Though there is training involved, it is purely unsupervised approach in which the model is built from the input data after representing it as a compressed model.

4.3 Simple Auto Encoder Based Anomaly Detection (SAE-AD) Algorithm

This algorithm is meant for taking the input process event log without considering probability distribution and its underlying parameters. It is the simple approach to have ANN based unsupervised learning to detect anomalies. It is deterministic in nature rather than probabilistic. It encodes given data efficiently and learn reconstruction of the same to original representation. In the process, it finds reconstruction error that is the basis for detecting anomalies. By design, it is capable of reducing dimensions in the dataset and learns how to ignore noise. In the process of encoding, it learns reducing dimensions and input data is compressed. The compressed form of the data is known as bottleneck with lowest number of dimensions. In the decoding process, the model learns reconstruction of original inputs. There is reconstruction loss method that finds the closeness of output data to the original data. A neural network (NN) is trained with back propagation to minimize loss in the reconstruction process. Loss is measured with MSE in this case. The training process is as in Algorithm 1.

Algorithm 1: Training Phase for SAE-AD

Input: Business process event log L={e1, e2, … e_n} to train

Output: encoder f_φ, decoder g_θ

φ, θ ← Initialize network parameters

repeat

Compute mean squared error

$$L_{(\varphi,\theta;x_i)}=\sum_i \| x_i - g_\theta(f_\varphi(x_i)) \|^2$$

φ, θ ← Update parameters by SGD

ISSN: 2005-4238 IJAST

until convergence of parameters (φ, θ)

Algorithm 1: Training phase for SAE-AD

As shown in Algorithm 1, the given event log dataset is used to train a model which is purely an unsupervised approach. This algorithm results in encoder and decoder for performing encoding and decoding operations on data later.

Algorithm 2: Simple Auto Encoder based Anomaly Detection

Input: Business process event log L={e1, e2, … e_n} to predict

Output: Reconstruction error $L_{(x,\hat{x})}$

φ, θ ← train network parameters with Algorithm 1

α ← set threshold based on training set

repeat

for i=1 to N do

Compute reconstruction error $L_{(x,\hat{x})}$

$$L_{(\varphi,\theta;x_i)}=\sum_i \| x_i - g_\theta(f_\varphi(x_i)) \|^2$$

If reconstruction error $L_{(x,\hat{x})}$> α **then**

x_i is anomalous

else

$x_i$is normal

end if

end for

Algorithm 2: Simple auto encoder based anomaly detection

As presented in Algorithm 2, it takes business process event log to predict anomalies. It finally produces reconstruction error which is in the form of MAE which is used to determine whether a data point is an anomaly. The underlying phenomenon of SAE-AD has similarity with non-linear Principal Component Analysis (PCA).

4.4 Probabilistic Auto Encoder Based Anomaly Detection (PAE-AD) Algorithm

Unlike simple auto encoder, the probabilistic auto encoder considers probability distribution in the input event log dataset. Its basic structure is similar to that of simple auto encoder. However, it is different from that of SAE. This model is meant for making a generative process model (Figure 4) which can cater to large scale datasets. It exhibits scalability and known for its latent representations. Moreover, it has control over modelling of latent distribution. This can improve representations it makes in the process of detecting anomalies.

ISSN: 2005-4238 IJAST

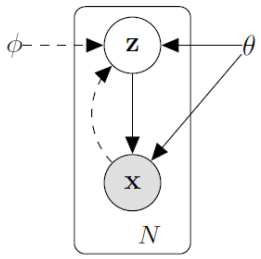

Figure 4. Generative process model of probabilistic auto encoder

As shown in Figure 4, it is the graphical model that represents a generative process model that systematically considers probabilistic parameters to be learned along with the generative model parameters.

Algorithm 3: Training Phase for PAE-AD

Input: Business process event log L={e1, e2, … e_n} to train

Output: encoder f_φ, decoder g_θ

φ, θ ← Initialize network parameters

repeat

X^M← random data from M data points

ϵ ← Random samples from noise distribution p(ϵ)

g ← $\nabla_{\theta,\varphi}\widetilde{\mathcal{L}}^M(\theta,\varphi;\ X^M,\epsilon)$(Gradients of minibatch estimator)

φ, θ ← Update parameters by SGD

until convergence of parameters (φ, θ)

Algorithm 3: Training phase for PAE-AD

Algorithm 4: Probabilistic Auto Encoder based Anomaly Detection

Input: Business process event log L={e1, e2, … e_n} to predict

Output: Reconstruction error $L_{(x,\hat{x})}$

φ, θ ← train network parameters with Algorithm 3

α ← set threshold based on training set

repeat

for i=1 to N do

Compute reconstruction error $L_{(x,\hat{x})}$

$L_{(\varphi,\theta;x_i)}=\sum_i \| x_i - g_\theta(f_\varphi(x_i)) \|^2$

if reconstruction error $L_{(x,\hat{x})}$> α **then**

x_i is anomalous

else

$x_i$is normal

ISSN: 2005-4238 IJAST

end if

end for

Algorithm 4: PAE based anomaly detection

Both the algorithms of anomaly detection use a measure to determine the anomalies in the event logs. The measure is known as Mean Absolute Error (MAE). It provides the average of actual difference between the actual and predicted observations. It is computed as in Eq. (1).

$$\text{MAE}=\frac{1}{n}\sum_{j=1}^{n}|y_j-\hat{y}_j| \quad (1)$$

In this paper MAE is used by the algorithms to determine the anomalies. A threshold is used to achieve this based on MAE value. MAE is considered a reconstruction error in this paper. If MAE value is more, it increases the probability of log entry being considered as anomaly.

5. Evaluation Methodology

The proposed algorithms are evaluated using a standard approach based on confusion matrix. Confusion matrix helps in deriving multiple metrics in machine learning approaches. They are used in process mining for prediction problems as well.

Table 1. Shows confusion matrix

	Ground Truth (Yes)	Ground Truth (No)
Result of SAE/PAE model (Yes)	True Positive (TP)	False Positive (FP)
Result of SAE/PAE model (No)	False Negative (FN)	True Negative (TN)

As shown in Table 1, confusion matrix provides different cases like TP, FP, FN and TN. These are used to derive performance metrics like precision, recall, F1 score, False Positive Rate (FPR) and True Positive Rate (TPR) as in Eq. (2), Eq. (3), Eq. (4), Eq. (5) and Eq. (6) respectively.

$$precision = \frac{TP}{TP+FP} \quad (2)$$

$$recall = \frac{TP}{TP+FN} \quad (3)$$

$$F_1 = 2.\frac{\text{precision} \cdot \text{recall}}{(\text{precision}+\text{recall})} \quad (4)$$

$$FPR = \frac{FP}{TN+FP} \quad (5)$$

$$TPR = \frac{TP}{TP+FN} \quad (6)$$

AUC-ROC isthe metric (used together) for measuring performance of a machine learning algorithm, especially in prediction problems. ROC stands for Receiver Operating Characteristics while AUC for Area Under Curve. The former is known as probability curve while the latter denotes degree of seperability. AUC reflects the capability of model to distinguish classes like normal or abnormal effectively.

ISSN: 2005-4238 IJAST

If AUC value is higher, the prediction model is better. ROC curve is based on TPR and FPR. FPR is represented in horizontal axis while TRP is on vertical axis. AUC value 1 means highest performance while 0 means worst performance. The value 0.5 indicates that the model has no separation capability. Therefore, AUC value beyond 0.5 indicates that there is some usefulness of the model.

6. Datasets

Real life process event logs are available in the form of Extensible Event Stream (XES) files. It is the standard for event logs. It has XML representation of data. OpenXES is the tool which is reference implementation of the XES standard in Java. It is preferred by IEEE task force on process mining under the name IEEE 1849-2016 XES. XES working group of IEEE developed this standard. The event logs are also available in the form of .CSV. In this paper four datasets are used for empirical study. They are known as BPI (Business Process Intelligence) Challenge 2015, BPI Challenge 2016, BPI Challenge 2017 and BPI Challenge 2018.

6.1 BPI Challenge 2015

This dataset is provided to public as part of BPI Challenge 2015. The real life log is collected from Dutch Financial Institute. It is available in [22]. It has 1308 cases and 262200 events across all cases. The log contains original data except few anonymizations. The log has five attributes. They are known as Case ID, Activity, Resource, Complete Timestamp, Variant, Variant Index, Amount_Req (case), Concept: Name and Lifecycle: Transition. The event logs are related to application processing for personal loan and overdraft in the financial organization which has global presence. The amount requested by customers is provided in the Amount_Req attribute which has same value for each case. The log reflects combination of three sub processes intertwined. The first letter of each activity denotes the sub process.

6.2 BPI Challenge 2016

This event log is collected from [23]. It is related to Volvo IT Belgium. The log has details of events associated with the system known as VINST. It is meant for managing problems and incidents in the company. The dataset has 1487 cases, 6661 events spreading across the cases and 14 attributes. The attributes include Case ID, activity, resource, complete timestamp, variant, variant index, concept: name, impact, lifecycle:transaction, org: group, org: role, organization country, organisation involved, product and resource country.

6.3 BPI Challenge 2017 and 2018

BPI Challenge 2017 dataset has 15000 cases, 120000 events and 8 attributes. The attributes include Case ID, activity, complete timestamp, variant, variant index, variable, variable_b and variable_c. The BPI Challenge 2018 dataset on the other hand contains 2000 cases, 28000 events and 5 attributes. The attributes are Case ID, activity, complete timestamp, variant and variant index.

ISSN: 2005-4238 IJAST

7. Experimental Results

Machine learning based algorithms that are part of the proposed framework for discovering anomalies from process event logs are implemented using Python's data science platform known as Anaconda. Results of two algorithms on four datasets are presented in this section.

7.1 Results of SAE-AD Algorithm

The observations of empirical study of SAE-AD algorithm are provided in terms of confusion matrix, reconstruction error and ROC curve.

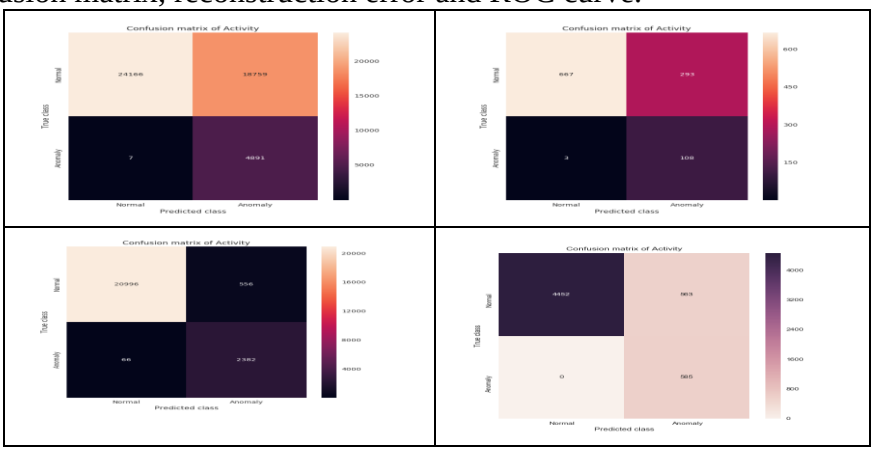

Figure 5. Confusion matrix of SAE-AD for activity of four datasets (top left: BPI Challenge 2015 dataset, top right: BPI Challenge 2016 dataset, bottom left: BPI Challenge 2017 dataset, bottom right: BPI Challenge 2018 dataset)

As shown in Figure 5, the confusion matrix with actual values or TP, FP, TN, FN for activity of all datasets is provided. For BPI Challenge 2015 dataset, TP value is 24166, FP 7, FN 18579 and TN 4891. For BPI Challenge 2016 dataset, TP value is 667, FP 3, FN 293 and TN 108. For BPI 2017 dataset, TP value is 20996, FP 66, FN 556 and TN 2382 while BPI 2018 dataset shows TP value is 4452, FP 0, FN 563 and TN 585. The results revealed that the confusion matrix of each dataset is different from others due to dataset size and its dynamics.

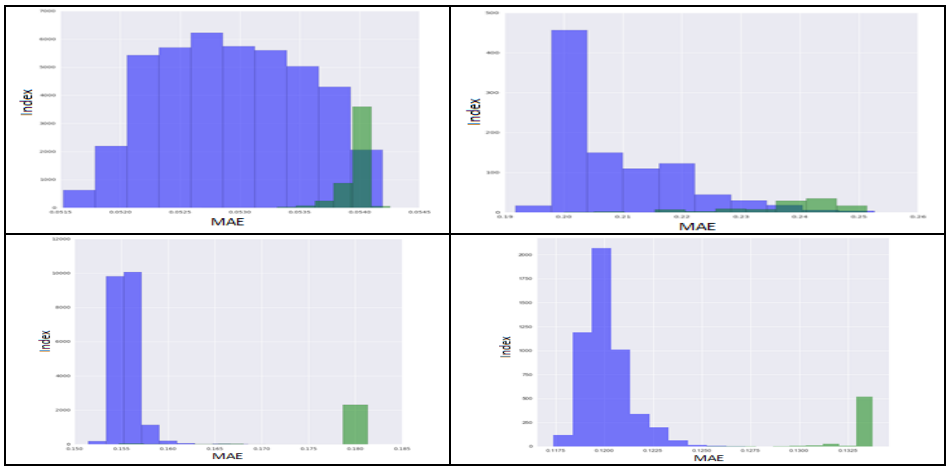

Figure 6. Reconstruction error of SAE-AD for activity of four datasets (top left: BPI Challenge 2015 dataset, top right: BPI Challenge 2016 dataset, bottom left: BPI Challenge 2017 dataset, bottom right: BPI Challenge 2018 dataset)

ISSN: 2005-4238 IJAST

As presented in Figure 6, the reconstruction error is for normal cases and anomaly cases. The results revealed that the reconstruction error is computed and presented for BPI 2015, BPI 2016, BPI 2017 and BPI 2018 datasets.

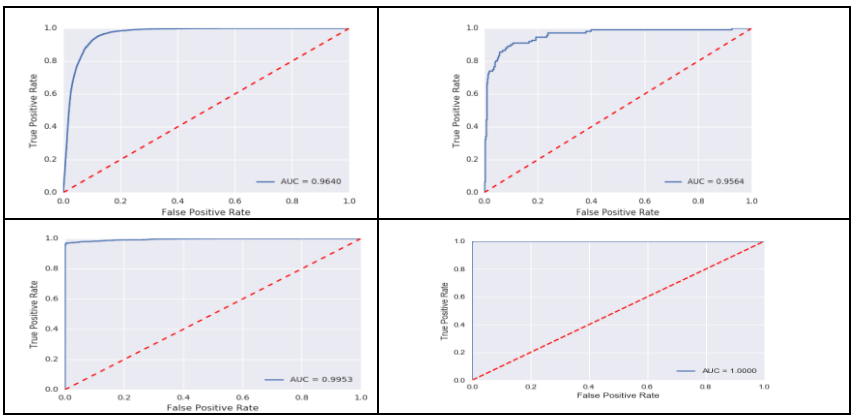

Figure 7. ROC curve of SAE-AD for activity of four datasets (top left: BPI Challenge 2015 dataset, top right: BPI Challenge 2016 dataset, bottom left: BPI Challenge 2017 dataset, bottom right: BPI Challenge 2018 dataset)

As presented in Figure 7, ROC curve is plotted with false and true positives represented in horizontal and vertical axes respectively. AUC reflects the capability of proposed algorithm to distinguish abnormal log entries from that of normal ones. The algorithm achieves 0.9640 for BPI 2015 dataset, 0.9564 for BPI 2016 dataset, 0.9953 for BPI 2017 dataset and 1.0 for BPI 2018 dataset. The results reflect significant capability of the algorithm in detecting anomalies.

7.2 Results of PAE-AD Algorithm

The observations of empirical study of PAE-AD algorithm are provided in terms of confusion matrix, reconstruction error and ROC curve.

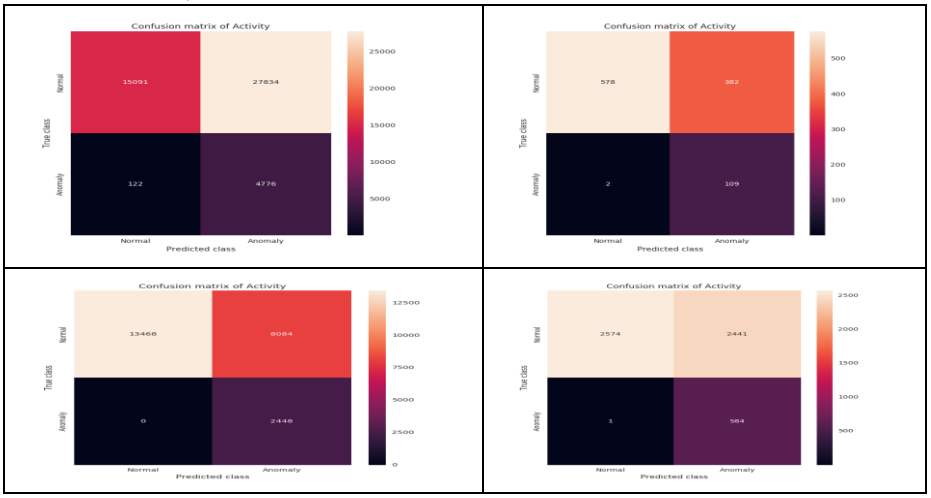

Figure 8. Confusion matrix of PAE-AD for activity of four datasets (top left: BPI Challenge 2015 dataset, top right: BPI Challenge 2016 dataset, bottom left: BPI Challenge 2017 dataset, bottom right: BPI Challenge 2018 dataset)

As shown in Figure 8, the confusion matrix with actual values or TP, FP, TN, FN for activity of all datasets is provided. For BPI 2015 dataset, TP value is 15091, FP 122, FN 27834 and TN 4776. For BPI 2016 dataset, TP value is 578, FP 2, FN

ISSN: 2005-4238 IJAST

382 and TN 109. For BPI 2017 dataset, TP value is 13468, FP 0, FN 8084 and TN 2448 while BPI 2018 dataset shows TP value is 2574, FP 1, FN 2441 and TN 584. The results revealed that the confusion matrix of each dataset is different from others due to dataset size and its dynamics.

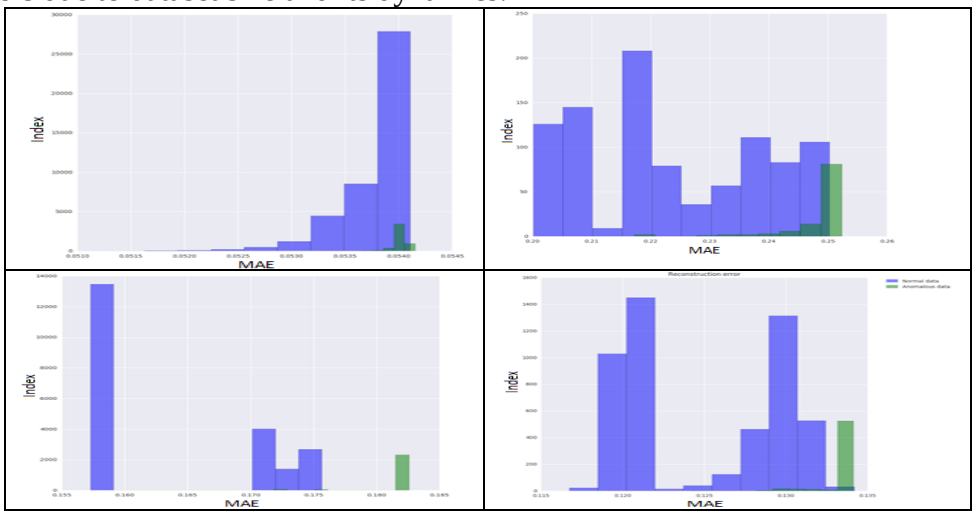

Figure 9. Reconstruction error of PAE-AD for activity of four datasets (top left: BPI Challenge 2015 dataset, top right: BPI Challenge 2016 dataset, bottom left: BPI Challenge 2017 dataset, bottom right: BPI Challenge 2018 dataset)

As presented in Figure 8, the reconstruction error is for normal cases and anomaly cases. The results revealed that the reconstruction error is computed and presented for BPI 2015, BPI 2016, BPI 2017 and BPI 2018 datasets.

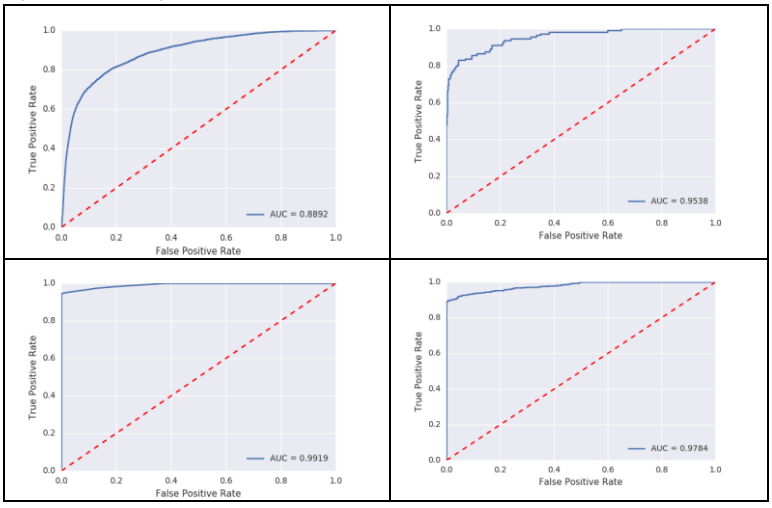

Figure 10. ROC curve of PAE-AD for activity of four datasets (top left: BPI Challenge 2015 dataset, top right: BPI Challenge 2016 dataset, bottom left: BPI Challenge 2017 dataset, bottom right: BPI Challenge 2018 dataset)

As presented in Figure 10, ROC curve is plotted with false and true positives represented in horizontal and vertical axes respectively. AUC reflects the capability of proposed algorithm to distinguish abnormal log entries from that of normal ones. The algorithm achieves 0.8892 for BPI 2015 dataset, 0.9538 for BPI 2016dataset, 0.9919 for BPI 2017 dataset and 0.9784 for BPI 2018 dataset. The results reflect significant capability of the algorithm in detecting anomalies.

ISSN: 2005-4238 IJAST

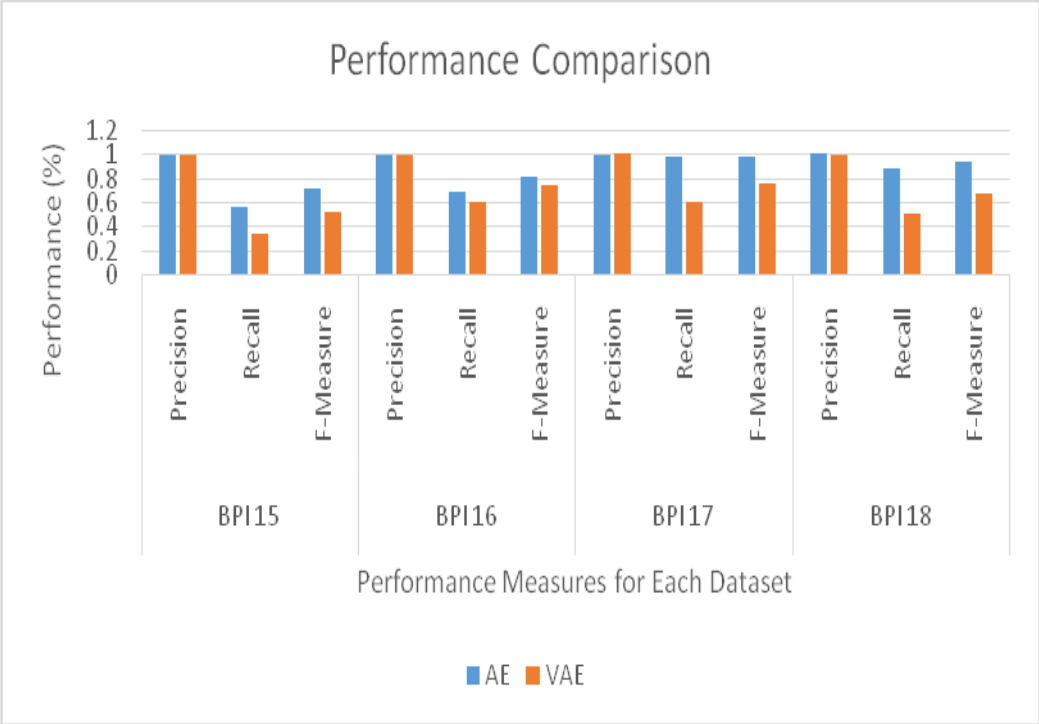

Figure 11. Performance comparison of algorithms

As presented in Figure 11, the performance of the algorithms is compared and evaluated with results of four datasets. The comparison is made in terms of precision, recall and F-measure. The precision of SAE-AD with BPI 2015 dataset is 0.99971 while PAE-AD showed 0.991981 for the same dataset. Recall for the SAE-AD and PAE-AD algorithms is 0.831676 and 0.351567 respectively. Similarly, F-measure is 0.907984 for SAE-AD while 0.519144 for PAE-AD. When the results are analysed in this fashion for other datasets also, there are important observations. PAE achieve precision as 1.0 for BPI 2017 dataset. Similarly, SAE achieved 1.0 precision for BPI 2018 dataset. The F-measure for all datasets with SAE-AD is higher than that of PAE-AD. The performance of the proposed algorithms shows the usefulness of the two approaches in building deterministic and probabilistic models to detect anomalies efficiently and enhance process logs.

8. Conclusion And Future Work

In this paper, we proposed a framework for efficient anomaly detection of business processes from event logs and process enhancement as well. The framework has two phases such as anomaly detection phase and event log reconstruction phase (process enhancement). The former is used to detect anomalies of business processes based on the log entries. This is carried out using an unsupervised ANN that learns compressing and encoding given event log dataset and then reconstruct data back from compressed representation to the original input. In the process, dimensions are reduced and the anomalies are detected. The latter on the other hand reconstructs event logs for the purpose of reconstructing event logs thus improving log quality improvement. It follows a multi-variate approach where two kinds of detectors one for anomalous activity and the other one for anomalous time. Mean Absolute Error (MAE) is used to

ISSN: 2005-4238 IJAST

measure reconstruction error with a threshold. When MAE greater than the threshold, it is known as anomaly. Two algorithms are proposed to realize efficient detection of anomalies. They are known as Simple Auto Encoder based Anomaly Detection (SAE-AD) and Probabilistic Auto Encoder based Anomaly Detection (PAE-AD). The SEA-AD does not consider probability distribution in the input data while the PAE-AD considers it and finds relevant parameters leading to generative process model. This makes PAE-AD scalable and feasible for large scale event logs. The performance of the two algorithms is evaluated in terms of precision, recall, F-measure and ROC curve. The empirical results revealed that the algorithms are efficient in detecting anomalies in event log and help in improving quality of it. These are much desired outcomes of process mining. In future, we intend to enhance the proposed framework with deep learning models.

References

[1] Fabio Bezerra and Jacques Wainer. Anomaly Detection Algorithms in business
Process Logs. p11-18.

[2] Kristof Bohmer and Stefanie Rinderle-Ma. (2016). Multi-Perspective Anomaly Detection in Business Process Execution Events. p1-18.

[3] Timo Nolle, Stefan Luettgen, Alexander Seeliger and Max (2019). BINet: Multi-perspective Business Process Anomaly Classification. Information Systems. p1-25.

[4] Kristof Bohmer and Stefanie Rinderle-Ma. (2017). Anomaly Detection in Business Process Runtime Behavior – Challenges and Limitations. p1-11.

[5] Andreas Rogge-Solti and Gjergji Kasneci. (2014). Temporal Anomaly Detection in Business Processes. Vienna University of Economics and Business. p1-16.

[6] Raffaele Conforti, Marcello La Rosa and Arthur H.M. ter Hofstede (2017). Filtering out Infrequent Behavior from Business Process Event Logs. A.H.M. ter Hofstede. p1-16.

[7] Kristof B¨ohmer and Stefanie Rinderle-Ma. (2017). Multi Instance Anomaly Detection in Business Process Executions. *springer*, p1-16.

[8] Ashish Sureka. (2015). Kernel Based Sequential Data Anomaly Detection in Business Process Event Logs. p1-4.

[9] Gabriel Marques Tavares,Victor G. Turrisi da Costa,Vinicius Eiji Martins,Paolo Ceravolo and Sylvio Barbon Jr. (2018). Anomaly Detection in Business Process based on Data Stream Mining. *ACM*. . (.), P120-127.

[10] Kristof Bohmer and Stefanie Rinderle-Ma. (2017). Multi Instance Anomaly Detection in Business Process Executions. University of Vienna. p1-16.

[11] Teemu Vartiainen. (2017). Analyzing event logs to discover process models and to detect anomalies in real-time. the degree of Master of Science in Technology. p1-61.

[12] Derek Lin and ChunSheng Fang. (2015). Historical Behavior Baseline Modeling And Anomaly Detection In Machine Generated End To End Event Log. EMC Corporation. p1-16

ISSN: 2005-4238 IJAST

[13] Andreas Wittenstein, Jim Lloyd, Mather and Mike Eynon. (2012). SystemAnd Method For Automatic Detection Of Anomialous Recurrent Behavor. Patent Application Publication. p1-48.
[14] Raffaele Conforti, Marcello La Rosa and Arthur H.M. ter Hofstede. (2018). Timestamp Repair for Business Process Event Logs. p1-14.
[15] Stephen Pauwels and Toon Calders. (2019). Detecting Anomalies in Hybrid Business Process Logs. APPLIED COMPUTING REVIEW. 19 (2), p18-30.
[16] Jennifer LeMond,Et al. (2015). Detecting Anomalous Accounts Using Event Logs. Microsoft Technology Licensing. p1-23.
[17] Titas SAVICKAS and Olegas VASILECAS. (2017). Decision Support Using Belief Network Constructed from Business Process Event Log. INFORMATICA. 28 (4), p687–701.
[18] Fabio Bezerra, Et.al. (2009). Anomaly Detection using Process Mining. Dep. of Mathematics and Computer Science. p1-13.
[19] Oh, D., & Yun, I. (2018). Residual Error Based Anomaly Detection Using Auto-Encoder in SMD Machine Sound. Sensors, 18(5), p1-14
[20] Xu, H., Feng, Et.al. Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications. Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18. p1-12.
[21] Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2017). Conditional Variational Autoencoder for Prediction and Feature Recovery Applied to Intrusion Detection in IoT. Sensors, 17(9), p1-17.
[22] https://www.win.tue.nl/bpi/doku.php?id=2016:challenge
[23] https://www.win.tue.nl/bpi/doku.php?id=2017:challenge

Authors

M. V Kamal, who could complete B.E in CSE from Gulbarga University, and M. Tech in Software Engineering from JNTU Hyderabad has been pursuing Ph.D in Data Mining from the JNT University, Hyderabad. He is having 18 years of experience in academia. He has published several papers in both National and International Papers and attended several National and International Conferences and organized Seminars etc. His area of interest includes Data Mining and Information Security

D Vasumathi, completed her B.Tech, and M.Tech from Jawaharlal Nehru Technological University Hyderabad. She did her Ph.D (Research) in the area of Data Mining from JNT University, Hyderabad. Presently she is working as Professor in Dept. of CSE, JNTUCEH and having more than 25 years of experience in teaching. She is a member for several professional bodies like CSI, IEEE and ISTE. She had presented and published several papers in National and

ISSN: 2005-4238 IJAST