An Event-Level Clustering Framework for Process Mining Using Common Sequential Rules

Zeeshan Tariq[1(✉)], Darryl Charles[1], Sally McClean[1], Ian McChesney[1], and Paul Taylor[2]

[1] School of Computing, Ulster University, Jordanstown, UK
{zeeshan,dk.charles,si.mcclean,ir.mcchesney}@ulster.ac.uk
[2] Applied Research, Ipswich, BT, UK
paul.n.taylor@bt.com

Abstract. Process mining techniques extract useful knowledge from event logs to analyse and improve the quality of process execution. However, size and complexity of the real-world event logs make it difficult to apply standard process mining techniques, thus process discovery results in spaghetti-like models which are difficult to analyse. Several event abstraction techniques are developed to group-up low-level activities into higher level activities, but abstraction ignores the low level critical process details in the real-world business scenarios. Also, trace clustering techniques have been extensively used in literature to cluster the processes executions which are homogeneous in nature, but event-level clustering is not yet considered for process mining. In this paper, a novel framework is proposed to identify event-level clusters in a business process log by decomposing into several sub-logs based upon the similarity of the sequences between events. Our technique provides clustering without abstraction of very large complex event logs. Proposed algorithm Common Events Identifier (CEI) is applied on a real-world telecommunication log and the results are compared with two well-known trace clustering techniques from the literature. Our results achieved high accuracy of clustering and improved the quality of resulting process models using the given size and complexity of the event log. We further demonstrated that the proposed techniques improved process discovery and conformance results for a given event log.

Keywords: Process mining · Process discovery · Trace clustering · Rule-based mining · Process analytics · Business processes · Sequence mining

1 Introduction

Collections of business processes enable organizations to operate efficiently and achieve their operational goals. A business process is a combination of activities performed by organizations to serve the needs of their internal or external

M. H. Miraz et al. (Eds.): iCETiC 2021, LNICST 395, pp. 147–160, 2021.
https://doi.org/10.1007/978-3-030-90016-8_10

customers [3]. These activities are recorded in the form of event logs and later utilized by businesses to assess the execution quality of their processes, resulting in identification of the possible improvement areas [19]. The complexity of organisation's business processes has also evolved with the growing diversity of the business environment in recent times. Execution of such processes lead to the generation of unstructured and variable event data which is prone to various inadequacies [23]. Proactive identification of these inefficiencies is critical in competitive business environments for maintaining alignment of an organisation's activities with its business goals.

Process mining is a set of tools and techniques used to discover, analyse and improve business processes [25]. Among other components, process discovery is one of the integral research area of the process mining domain. Control flow models of the process such as petri nets, helps in analysing the way process is executed and evaluate the dispensaries between ideal process model and generated event log [20]. Discovered models from the real world events are generally unstructured and exhibit a spaghetti-like pattern having low model quality [20]. Things become further complicated where the number of events in the process is extensively large and the outcome of the process directly affects the financial aspect of the organisation, such as customer churn out rate in telecommunication sector. A recent focus of research is in development of the techniques for trace clustering for identifying clusters in the large event logs with lowest level of events recorded, results in spaghetti-like process models. Clustering techniques are mostly based on the similarity of traces within an event log [16]. Generally, techniques focused on trace clustering neglect the business perceptive of the process while dealing with real-world logs [21].

Event logs may contain very low level of process details, event-abstraction techniques are developed to group the low-level events into high level events [17], but these techniques tend to ignore the activities which are meaningful for analysis at lower abstraction level. When a real-world process execution is recorded, there may be similar activities existing in the log which reflects that there exists a common behaviour in the process execution which is reflected in all of the recorded traces. The importance of the identification of this common behaviour is elevated where number of events per trace is exponentially large and business process is composed of several sub-processes. Also, such discovery eliminates possible performance overheads due to presence of such "always occurring" similar events in a competitive business environment. Our approach is presented in Fig. 1, presenting that raw complex log with large number of events is decomposed into several sub-logs. In Fig. 1, sub-figure (a) shows a spaghetti-like process model of the services diagnostic process, for the customers at a telecommunication firm. The event log contains several subcategories of traces and clusters of events which are common in all the traces. When common events are identified and segregated from the log, several sub-logs are created as a result of this event-level segmentation. Sub-figures (b) & (c) are the segments of the traces with common events while sub-figure (d) presents the remaining portion

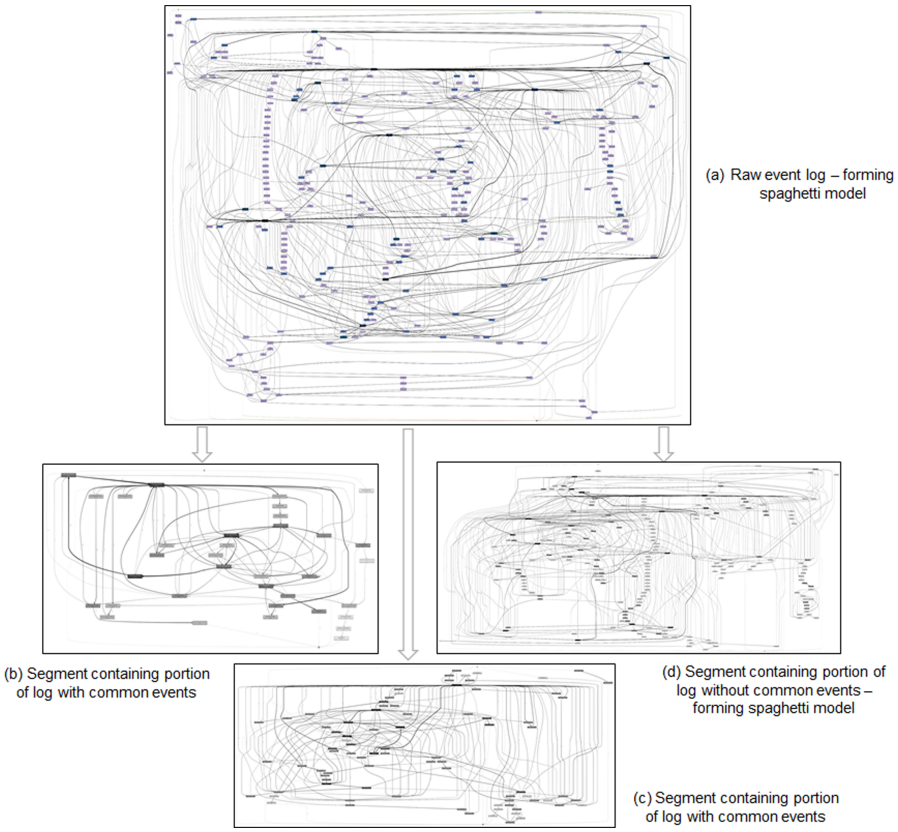


Fig. 1. Event log segmentation as a result of common events identification.

of the log with events with similarity in association rules higher than the defined threshold.

Our contributions in this paper are summarised as follows:

- We investigated the event-level clustering for the complex event logs where number of events are exponentially large. Solutions such as trace clustering and event abstraction, provides decomposition of an event log but clustering of low level abstracted events are not considered.
- A novel approach for the clustering of the event log is presented for identification of the similarity in the events through the sequential rule-based mining technique. An algorithm Common Events Identifier (CEI) is proposed to identify the portion of the process log having similar sequential rules, referred in this paper as Common Event Segment (CES).
- We conducted several experiments to evaluate the proposed technique on a real-world telecommunication data. We also compared the results with other well-known trace clustering techniques from the literature, ActiTraC proposed by Song et al. [7] and NoHiC by Tariq et al. [21].

In Sect. 2, we discussed several techniques from the related research, Sect. 3 provides an overview of the proposed framework, Sect. 4 demonstrates the outcome of the experiments and related discussions. Finally, Conclusion and Future Work are presented in Sect. 5.

2 Related Work

Process mining techniques are developed to discover process models from event logs which assists in analysing and enhancing the quality of process execution [20]. Process mining domain bridges the gap and provides ground for analysis of event logs through data mining techniques, as mentioned in [2], and business process analytics. Trace clustering is one of a key research area in process mining in which a complex event log is decomposed by discovering the clusters based on similarity of cases/traces [8,16]. In [16] a trace clustering technique has been proposed to identify frequent sequence patterns to discover the clusters in healthcare dataset using domain experts input about subcategories of the main process trunk. Several clustered traces are then ranked on the basis of their sequence patterns to find most frequent traces. Work of [16] presents understandable trace clustering, similar to our work, instead we used rule based mining which is more affective in terms of implementation and understanding of underlying event-sequences. Furthermore, our work is based on self-identification of difference subcategories of business process based on the difference in the sequential rules. In another work on trace clustering, a tree-based trace clustering technique is proposed in [6] where the process is clustered using a iterative approach using a DWS (Disjunctive Workflow Schema) algorithm. A general trace clustering techniques is proposed in [5] providing an environment to perform broad range of process centric analysis to indicate correlation in several process characteristics, such as control-flow, data-flow time resources and conformance. Abstraction techniques analyses event logs with very low level of abstraction by converting low-event events into high level events. Authors of [17] proposed a methodology to initially discover the Local Process Models(LPM) at higher abstraction level of events. Authors showed that the composite of LPM models with high-level activities resulted in improved fitness and precision of discovered model.

Rule-base mining is used for finding interesting sequence patterns among the events through extraction of sequential patterns [14]. In [21], authors presented rule-based mining for segregation of different classes in business process data, in order to support clustering of cases based on agglomerative approach. A rule based algorithm is proposed in [9] to address the problem of missing unique identifiers in the event log. Proposed algorithm in [9] use varying threshold of similarity between events to combine the events in a group. We also use rule based mining in this paper but our focus is to group those events which are in the form of a sequence thus not disturbing the underlying flow of the business process.

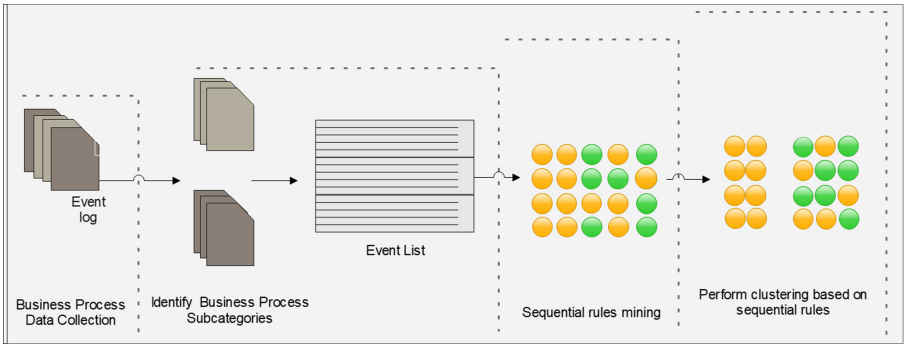


Fig. 2. Proposed framework for the identification of common events cluster

3 Methodology

This paper presents a solution for identification of common events existing in the lengthy heterogeneous business processes. These events are present in the logs as a cluster of similar activities performed during all the traces. Business processes in modern corporations encompassing a variety of process subcategories. Subcategories of the process are referred to as independent groups of the instances which are a part of a business process. Organizations allow these hybrid categories of instances in a business process to manage diverse business scenarios in a highly competitive business environment. Event logs contain these subcategories as different classes of traces. Irrespective of occurrence and duration,it is possible that different subcategories of the business process instances contain several such events which are common across all subcategories of the business process.

We project that identification of these clusters does two main roles, (i) decomposing the extensively large logs into sub-logs, thus making the discovery phase more effective, (ii) common events in logs may be serious processing overhead and should be considered for several process enhancement measures, such as compression, time reduction or replacing them with automated systems.

3.1 Framework for Common Events Identification

The framework proposed in this paper is composed of four stages starting with the collection of raw event log, presented in Fig. 2. We discuss stages of our framework through a case study of a call centre process at BT, one of the leading telecom firms in the UK. Details of each stage are as follows:

Event Log. Events are the activity performed by a resource within a scope of a process instance at the given time stamp [5]. We used a real-world event log of a call centre setup at BT. Customers contact the organisation (BT) for the solution of their service-related queries through a semi-automated chat service, termed

Table 1. Summary of the customer diagnostic process log

Dataset	Traces	Min length of trace	Max length of trace	Total number of events	Mean classes per case
CDP	2000	5	660	489804	169
Subcategory A	1000	13	660	276635	178
Subcategory B	1000	5	302	213169	159

as a Customer Diagnostic Process (CDP). Customer diagnostics are performed through a series of system generated questions. A similar scenario of sequential questions is presented in [18]. The flow of the process for CDP is presented in Fig. 3. Each instance is labelled with a unique Case_ID. Event details are abstracted with numbers IDs for privacy regulations. Initial preprocessing of the data is performed, which includes the removal of duplicate instances and exclusion of uncompleted logs. Details of the CDP log is mentioned in Table 1.

Identification of Business Process Subcategories. Event log from organizational information systems generally include instances belonging to several subcategories. These subcategories can be identified through several ways, such as by exploring the broad characteristics of process instances or getting direct information from domain experts. In our dataset, customers interacting with CDP belong to two subcategories, Broadband customers and PSTN customers. Table 1 presents the details of these subcategories.

Event List. The event list is a tabular view of cases with their associated events, represented as L. The first column of L represents the case unique_id, and the remaining columns present the events performed in sequential order. Each row of L represents the execution of a complete instance. For simplicity, we are only considering traces where starting event for all the cases is restricted to single class.

Sequential Rules Mining. This stage generates the sequential rules for the given events under certain performance metrics using the Apriori algorithm. Sequential rule-based mining technique is used to identify the frequent subsequences of events in all the cases of the considered portion of the log. Rules are extracted for each of the subcategories of the CDP through the Apriori algorithm [1] using R. Rules are extracted using the mandatory performance metrics

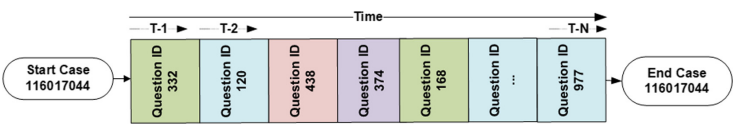


Fig. 3. Sample diagnostic process with labelled events

such as support, confidence and lift. Apriori learns the rule in the form of ⇒, which means that every time X (event) appeared in the given unique process then Y (event) appeared at least once in that given sequence and X is always followed by Y.
To validate the performance of the rules, the following three metrics are used:

- **Support:** A measure of the applicability of a rule used. Let X and Y are two events, N_X be the number of instances for which X holds & N_Y be the number of instances for which Y holds then *Support* of a rule X ⇒ Y is the proportion of cases which holds for both X and Y. Higher support shows more applicability of the approach. If N is total number of cases then Support of a rule is defined as (1).

$$Support(X \Rightarrow Y) = N_{X \wedge Y}/N \tag{1}$$

- **Confidence:** It is worth of the rule with respect to the reliability within the given dataset. A value of the confidence closest to 1 qualifies extent of the given rule's worth to be represented with maximum confidence. Confidence of a rule is defined as (2).

$$Confidence(X \Rightarrow Y) = N_{X \wedge Y}/N_X \tag{2}$$

- **Lift:** It is the measure of positive correlation between X and Y. A value of lift closer to 1 indicates that X and Y are more positively correlated & independent to each other. Lift of a rule is defined as (3).

$$Lift(X \Rightarrow Y) = N_{X \wedge Y} N/N_X N_Y \tag{3}$$

Sample of sequential rules generated for CDP are presented in Table 2. The process of identifying the frequent subsequences using the TraMineR package [12] implemented in the R tool is; First the Time Stamped Event (TSE) format of event log is created along with event sequence objects. Then frequent subsequences among the events are generated and sequential rules are extracted based on threshold values of $Support$, $Confidence$ and $Lift$. Finally, discriminating sub-sequences are identified, which shows level of inheritance of rules in a defined cohort of data. Frequent subsequences of events occurring in the traces of subcategory-A are not necessarily the same as those which exists in the subcategory-B.

Algorithm for Common Events Identification. We proposed an algorithm Common Events Identifier (CEI) for the discovery of clusters with common events in a given process log. CEI consists of 4 steps, where Step-1 concerns with the preparation of the process log. Consider a process log α having events e forming an event list L. In Step-2, L is traversed with the single step increment, given as window size n, and sequential rules for the events corresponding to L are extracted. i is pointer for the starting column of the L. The value of window n increases with every iteration following the sliding window principle. In

Table 2. Sample rules generated from the CDP process log with the values of *Support*, *Confidence* and *Lift*

	Rules	Support	Confidence	Lift
1	{1332} ⇒ {1442}	0.8131	0.9737	1.0124
2	{1442} ⇒ {108}	0.6949	0.8942	1.0928
3	{1005} ⇒ {1004}	0.6932	0.8920	1.0884
4	{1005} ⇒ {1002}	0.6932	0.8920	1.0884
5	{1198} ⇒ {1887}	0.6876	0.9833	1.3772
6	{103} ⇒ {1093}	0.6837	0.9857	1.4139
7	{21} ⇒ {100}	0.6837	0.9808	1.4139
8	{11} ⇒ {220}	0.6837	0.9808	1.4139

Step-3, we considered the generated sequential rules at each iteration for evaluation of the Pearson residual correlation [12] using Chi-square test. Correlation is detected between different cohorts of rules generated by CBA algorithm. A fixed threshold of 10% correlation is considered to distinguish the commonality of rules between the cohorts. We kept the low value of threshold to make sure that slightest difference in rules between cohort is effectively detected. Finally, Step-4 identifies the columns in the event list which are a part of the Common Events Segment (CES). Discrimination residual represents the frequency of occurrence of any specific subsequence.

All test runs are performed on a desktop computer with an Intel Core-i5-8th Gen processor running at 1.80 GHz, 16 GB of RAM, Windows 10 Enterprise (64-bit), and a 64-bit version of Java 8 with 8 GB of RAM assigned to the Java virtual machine. Time to run single iteration varied from 0.7 s to maximum of 33 mins depending upon the length of trace.

Dissimilarity in the cohorts is identified as a Pearson residual value P between -1 and $+1$. If P increases to the certain threshold ($P = 0.1$) all e in the list L till γ will be marked as CES. Algorithm stops when task list L reaches to an end.

Iteration of CEI on CDP Event Log. Findings from the iterations of CEI on customer diagnostic process log are as follows:
1st iteration: $i = 1$, $n = 2$, $P = 0$,where i is indicating the 1st column of the event list L, n is starting from 2 as to consider first two columns of events. Support minimum threshold is kept at default value of 0.2.
2nd iteration: $i = 1$, $n = 3$, $P = 0$. Several new sequential rules are generated during the 2nd iteration of CEI but all these rules are mutually common between Broadband and PSTN cases.
3rd iteration: At the end of the 3rd iteration, $i = 1$, $n = 3$, $P = 0$. The value of n keeps on increasing as a sliding window grows until the value of P increase from the set threshold.

Algorithm 1: High-level pseudo-code description of the Common Events Identifier (CEI) Algorithm

Step-1: For an event log α
i=1(start of α);
n=2 (initializing with 2nd column in the event list);
e=Set of events in each column between i and n;
Step-2: If (e=NULL)
Exit,
else Generate sequential rules for e;
Step-3: Evaluate the Pearson residual correlation P of the Chi-square test;
if *(correlation value of* $P \neq 0$*)* **then**
    $max = n$-1;
    Cs (start of cluster) = event in the list at position i;
    Ce (end of cluster) = event in the list at position max;
    $i = n$;
    $n = i$+1;
else
    until $n \neq$ End of event list;
    $n = n$+1;
end
Step-4: *Common Events Cluster* = Events between Cs and Ce
Go to ***Step-2*** (for discovery of further clusters)

Similarly, algorithms traverse the event list L and compare the resulting rules with minimum threshold of correlation.

CES marking As CEI continues to traverse L, at the end of 106th iteration the value of $n = 107$, $P = +0.1$. Results from the cohort analysis show the discrimination is identified, as the value of P is > 0.1. This presents that a set of sequence rules now exists within n columns of the event list which discriminate between cohorts. Portion of the log with common events is identified as CES.

4 Results and Discussion

This section presents the results of the proposed framework. The quality of the clustering process is evaluated and later we showed the accuracy of the log segments (sub-logs) generated as a result of clustering. We performed the quality assessment of identified clusters in comparison with other well-known techniques from the literature. In this section, firstly, the weighted Shannon Entropy [15] to measure the change in entropy of the CES as compared to the rest of the process log. Secondly, Classification based on Association (CBA) [13] technique to measure the quality of the discovered clusters. We presented CBA results with performance metrics, such as accuracy, precision, sensitivity and the F-measure.

4.1 Measuring the Quality of Clusters

To measure the quality of the generated clusters, we used two methods. The first method is the weighted Shannon entropy [10], to measure the entropy of the event list L. The second method is to measure the prediction accuracy of the discovered clusters using Classification based on Association rules (CBA) [11].

Entropy Change Between the Event. Shannon's Information Entropy is the average rate at which information is produced by a stochastic source of data [15]. The entropy H is calculated for list L as value of randomness between activities of each column. Entropy H is a probability p of an activity i appearing in the given column of a event list. The formula of Shannon entropy is given by (4):

$$H = \sum_{i=1}^{m} p_i log_2 p_i \tag{4}$$

Figure 4 shows the graph of the weighted Shannon's entropy calculated for columns of the event list L. Change in the entropy Entropy H reflects the change in randomness as observed in Fig. 4. Increase in the Entropy H is an indicator for the decluttering of events in the log. Portion of the log with minimum randomness is the segment with common events, thus event log is divided into two sub-logs, one with common events and other with remaining events.

Clustering Accuracy. As a result of clustering, process lag is decomposed into several segments. We compared the association rules between different segments of the process log through the CBA technique [13] implemented in R. Results are presented in the Table 3 which shows that the overall accuracy of the classification of segments is above 90% for all the traces where minimum support for the generated rules is greater than or equal to 30%. The value of confidence is set as a default in CBA (80%). On average, 95% accuracy is achieved with two variations of testing and training data sets, which suggests that clusters are well segregated. An average sensitivity of around 90% presents the accuracy of the discrimation of rules between events of different segments. The F1 measure varies between 91.7% and 98.8% for all support and testing percentage scenarios presenting high precision and recall values.

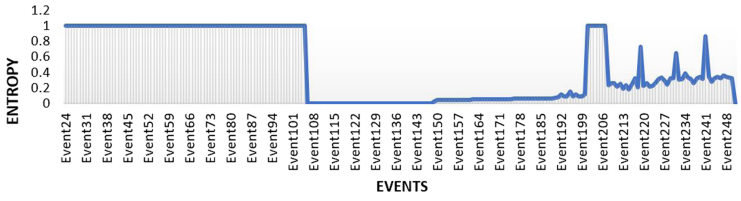


Fig. 4. Shannon Entropy calculated for the events in the process log

Table 3. Accuracy of the identified clusters

Parameters			Results		
Support	Training data	Testing data	Accuracy	Sensitivity	F1 Measure
30%	80%	20%	97.22%	94.00%	97.14%
35%	80%	20%	98.12%	96.20%	98.00%
40%	80%	20%	94.40%	88.80%	94.11%
45%	80%	20%	94.40%	88.80%	94.11%
30%	85%	15%	98.80%	97.60%	98.80%
35%	85%	15%	92.35%	84.71%	91.70%
40%	85%	15%	96.47%	87.70%	91.00%
45%	85%	15%	95.88%	91.76%	95.00%

4.2 Comparison with Other Trace Clustering Techniques

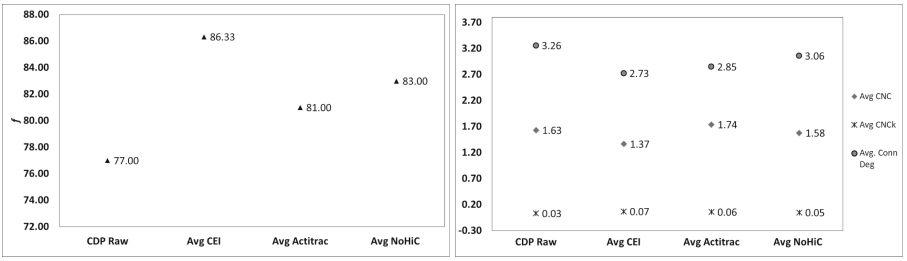
To evaluate the impact of our technique on the process discovery, we generated process models for all the sub-logs using heuristic miner three quality criterion of process models, as mentioned in [24], fitness *f*, Coefficient of connectivity(CNC) & Coefficient of network complexity (CNC_k), are compared with the clustering results of other techniques from the literature. Results from the Algorithm CEI is compared with two clustering techniques from the literature, Trace clustering (ActiTraC) [7] and NoHiC [21].

For ActiTraC, we kept the default settings as implemented in the ProM 6.10 plugin *ActiTraC Clustering*. We compared the results of clustering based on 3 quality criteria of process discovery which are, fitness of model (f), Coefficient of connectivity(CNC), Avg Degree of Connectivity(CD), and Coefficient of network complexity (CNC_k) detailed in [24]. Fitness *fitness* gauge the quality of the process model by measuring the events mismatch when the event log is replayed with the discovered process model [4]. A process model is said to have perfect *fitness* if it allows replay of all the traces in the process log at the given petri net model. Equation (5) presents the fitness f of the process log σ on petri net η.

$$f(\sigma, \eta) = 1/2(1 - m/c) + 1/2(1 - r/p) \tag{5}$$

where m = missing tokens, c = number of consumed tokens, r = number of tokens remaining after replay,p = total number of tokens produced.

Figure 5 shows two comparison plots. In, Fig. 5a overall fitness of raw CDP log is 77%, which is elevated to 86% with the segregation of log into common events clusters and remaining portion of trace. This increase in fitness is due to the segregation of events between different segments/sub-logs. Portion of the log with common events results with higher fitness value. NoHiC and ActiTrac discovered 4 clusters each, but average fitness of models is yet slightly lower than CEI, 83% and 81% respectively. Decrease in CNC from 1.63 to 1.36 by CEI is shown in Fig. 5b which evident the impact on complexity of the log. There



(a) Comparison of average fitness f (b) Comparison of CNC, CD & CNC_k

Fig. 5. CDP raw is the process log without any clustering. CEI outperformed for several criterion including fitness Coefficient of Connectivity(CNC), Avg Degree of Connectivity(CD) & Coefficient of Network Complexity(CNC_k)

is a slight increase in the value of CNC_k as those segments emerged from the log which got high graph network complexity, depicting that events with more randomness are segregated in a sub-log. Raw log has CD of 3.26 presenting high level of connectivity in comparison to the average fragmented logs. Average of CEI has CD lowest among all which is 2.73 but this is an average of fragments' CD ranging from 2.0 (for cluster with common events) to 4.4 (most fuzzy part of the log). Yet, lowest average CD presents the better precision of the resulting process models.

5 Conclusion

Event data extracted from real-life business processes is very large and highly unstructured. Discovery of such processes lead to the complicated patterns which are difficult to understand with traditional process mining techniques. In this paper we proposed a framework to simplify the event log by decomposing it into manageable segments. Our technique discovers the clusters of the common events within a business process log thus allowing large log to be fragmented into easy manageable portions. We demonstrated through real-world case study that our technique improved the process discovery. We achieved high accuracy of clustering using our proposed algorithm CEI and compared the results with other techniques from the literature. We also presented the accuracy of clusters through entropy calculation of event log and cohort-based analysis using CBA algorithm. For future, we will extend our work in two dimensions (i) conformance analysis of the identified log segments, and (ii) incorporating further complexities in the event log, such as, multiple start/finish classes and trace misalignments.

Acknowledgment. This research is supported by the BTIIC (BT Ireland Innovation Centre) project, funded by BT and Invest Northern Ireland.

References

1. Aggarwal, C.C., Bhuiyan, M.A., Hasan, M.A.: Frequent pattern mining algorithms: a survey. In: Aggarwal, C.C., Han, J. (eds.) Frequent Pattern Mining, pp. 19–64. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07821-2_2
2. Ashraf, N., Ahmad, W., Ashraf, R.: A comparative study of data mining algorithms for high detection rate in intrusion detection system. Ann. Emerg. Technol. Comput. (AETiC), pp. 2516–0281 (2018). Print ISSN: 2516–0281
3. Borgianni, Y., Cascini, G., Rotini, F.: Business process reengineering driven by customer value: a support for undertaking decisions under uncertainty conditions. Comput. Ind. **68**, 132–147 (2015)
4. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P., et al.: On the role of fitness, precision, generalization and simplicity in process discovery. In: Meersman, R. (ed.) OTM 2012. LNCS, vol. 7565, pp. 305–322. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33606-5_19
5. De Leoni, M., van der Aalst, W.M., Dees, M.: A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. Inf. Syst. **56**, 235–257 (2016)
6. de Medeiros, A.K.A., Guzzo, A., Greco, G., van der Aalst, W.M.P., Weijters, A.J.M.M., van Dongen, B.F., Saccà, D.: Process mining based on clustering: a quest for precision. In: ter Hofstede, A., Benatallah, B., Paik, H.-Y. (eds.) BPM 2007. LNCS, vol. 4928, pp. 17–29. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78238-4_4
7. De Weerdt, J., Vanden Broucke, S., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. IEEE Trans. Knowl. Data Eng. **25**(12), 2708–2720 (2013)
8. Delias, P., Doumpos, M., Grigoroudis, E., Matsatsinis, N.: A non-compensatory approach for trace clustering. Int. Trans. Oper. Res. **26**(5), 1828–1846 (2019)
9. Djedović, A., Karabegović, A., Žunić, E., Alić, D.: A rule based events correlation algorithm for process mining. In: Avdaković, S., Mujčić, A., Mujezinović, A., Uzunović, T., Volić, I. (eds.) IAT 2019. LNNS, vol. 83, pp. 587–605. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-24986-1_47
10. Eskov, V., Eskov, V., Vochmina, Y.V., Gorbunov, D., Ilyashenko, L.: Shannon entropy in the research on stationary regimes and the evolution of complexity. Mosc. Univ. Phys. Bull. **72**(3), 309–317 (2017). https://doi.org/10.3103/S0027134917030067
11. Filip, J., Kliegr, T.: Classification based on associations (CBA)-a performance analysis. Tech. rep, EasyChair (2018)
12. Gabadinho, A., Ritschard, G., Studer, M., Mueller, N.: Mining sequence data in r with the traminer package. University of Geneva, A User's Guide. Department of Econometrics and Laboratory of Demography (2011)
13. Hahsler, M., Johnson, I., Kliegr, T., Kucha, J.: Associative classification in r: arc, arulesCBA, and rCBA. R J. **9**(2) (2019)
14. Lim, A.H., Lee, C.S.: Processing online analytics with classification and association rule mining. Knowl.-Based Syst. **23**(3), 248–255 (2010)
15. Lin, J.: Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theory **37**(1), 145–151 (1991)
16. Lu, X., Tabatabaei, S.A., Hoogendoorn, M., Reijers, H.A.: Trace clustering on very large event data in healthcare using frequent sequence patterns. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNCS, vol.

11675, pp. 198–215. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26619-6_14
17. Mannhardt, F., Tax, N.: Unsupervised event abstraction using pattern abstraction and local process models. arXiv preprint arXiv:1704.03520 (2017)
18. Onik, M.M.H., Al-Zaben, N., Hoo, H.P., Kim, C.S.: A novel approach for network attack classification based on sequential questions. Ann. Emerg. Technol. Comput. (AETiC), pp. 1–14 (2018). Print ISSN:2516–0281
19. Rojas, E., Munoz-Gama, J., Sepúlveda, M., Capurro, D.: Process mining in healthcare: a literature review. J. Biomed. Inform. **61**, 224–236 (2016)
20. Rudnitckaia, J.: Process mining: Data science in action, pp. 1–11. University of Technology, Faculty of Information Technology pp (2016)
21. Tariq, Z., Khan, N., Charles, D., McClean, S., McChesney, I., Taylor, P.: Understanding contrail business processes through hierarchical clustering: a multi-stage framework. Algorithms **13**(10), 244 (2020)
22. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Event abstraction for process mining using supervised learning techniques. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2016. LNNS, vol. 15, pp. 251–269. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-56994-9_18
23. Taylor, P., Leida, M., Majeed, B.: Case study in process mining in a multinational enterprise. In: Aberer, K., Damiani, E., Dillon, T. (eds.) SIMPDA 2011. LNBIP, vol. 116, pp. 134–153. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34044-4_8
24. Thaler, T., Ternis, S.F., Fettke, P., Loos, P.: A comparative analysis of process instance cluster techniques. Wirtschaftsinformatik **2015**, 423–437 (2015)
25. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) Business Process Management Workshops, BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_19