UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science

Mahmoud Kamel Shoush

Improved Classifier Training Methods for Predictive Process Monitoring

Master Thesis (30 ECTS)

Supervisor: Prof. Marlon Dumas

TARTU, 2020

Abstract

Recently, there has been numerous studies on the use of machine learning (ML) methods for business enhancement across different areas. Organizations are in need to improve their business process performance by utilizing predictive models for monitoring ongoing business cases. Predictive process monitoring (PPM) tackles this problem by forecasting the behaviour, execution, and outcome of business processes at runtime. PPM approaches take an event log (i.e. a collection of completed cases) as input and utilize ML methods to train models to predict the future state of a given case, and to answer questions such as: Will a loan application will be approved or declined (i.e. final outcome)? What is the next event given the previous events? Or what is the remaining time until the end of the case? A specific, family of approaches of PPM, known as outcome-oriented PPM, focuses on predicting whether or not a case will end with an expected outcome or not. An outcome-oriented PPM framework is expected to form precise predictions in the early execution stages to decide if the system worker should take part and get involved or not and to avoid unexpected outcomes. In this setting, this thesis addresses the question of how to improve the predictive process monitoring of business process outcomes. To answer this question, we propose three different enhancements to the currently existing approaches that have been introduced in the literature. The proposed enhancements are evaluated using a benchmark covering 20 prediction tasks that come from different real-life event logs. Empirical results confirm that our proposed approaches deliver significant improvements relative to existing PPM techniques in terms of accuracy.

Keywords— Predictive process monitoring, Process Mining, Machine Learning

CERCS: P170 - Computer science, numerical analysis, systems, control

Pealkiri: Täiustatud masinõppemeetodid äriprotsesside ennustava seireks

Lühikokkuvõte: Hiljuti on tehtud mitmeid uuringuid masinõppemeetodite (ML) kasutamises ettevõtluse edendamiseks erinevates valdkondades. Organisatsioonidel on tarvis parandada oma äriprotsesside toimivust, kasutades ennustusmudeleid protsesside jooksvaks seireks. Protsesside ennustav seire (PPM) aitab selle probleemiga toime tulla, prognoosides äriprotsesside käitumist, täitmist ja tulemusi tööajas. PPM-lähenemised võtavad sisendina sündmuste logi (s.o täielike ärijuhtumite kogumi) ja kasutavad ML-meetodeid, et õpetada ennustavaid mudeleid konkreetse äriprotsessi tulevase oleku ennustamiseks. Käimasoleva juhtumi ennustatav tulevane seisund võib varieeruda sõltuvalt probleemi eesmärgist, vastates näiteks küsimustele: kas laenutaotlus võetakse vastu või lükatakse tagasi (s.o lõpptulemus), milline on järgmine sündmus, võttes arvesse varasemaid sündmusi? Või mis on lõpuni jäänud aeg? On olemas PPM-i lähenemisviiside perekond, mida tuntakse kui tulemustele orienteeritud PPM-i ja mis keskendub prognoosimisele, kas ärijuhtum lõpeb oodatava tulemusega või mitte. Selliselt tulemustele orienteeritud PPM-raamistikult oodatakse varajastes täitmisetappides täpseid ennustusi, et otsustada, kas süsteemitöötaja peaks sekkuma või mitte, et vältida ootamatuid tulemusi. Eelnevat arvestades käsitletakse käesolevas lõputöös küsimust, kuidas parandada äriprotsesside tulemusi ennustavate protsesside jälgimist. Sellele küsimusele vastamiseks pakume välja kolm erinevat täiendust kirjanduses kasutusele võetud olemasolevatele lähenemisviisidele. Lisaks antakse käesolevas töös ülevaade pakutud meetodite võrdlevast uurimuslikust hinnangust, kasutades võrdlusalust, mis hõlmab 20 ennustuslesannet, mis pärinevad erinevatest reaalajas aset leidnud sündmuste logidest. Empiirilised tulemused kinnitavad, et meie väljapakutud tehnikad parandavad märkimisvrselt praeguste PPM-meetodite täpsust.

Võtmesõnad: Ennustav protsessiseire, protsessikaeve, masinõpe

CERCS: P170 - Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Contents

List of Figures

List of Tables

Acknowledgements

I would like to appreciate my thesis supervisor Prof. Dr Marlon Dumas, for the supervision, support, and guidance he has given to me during a whole academic year. It would not have been achievable for me to defend if it was not by his advice and help. He always had the patience to listen to my thoughts, give productive feedback and, direct me in the correct path whenever I wanted it the most. I am additionally extremely grateful for the various opportunities he gave me, where I obtained several research experiences, learned around several exciting topics in the area of Process Mining.

Besides my supervisor, I would like to thank the University of Tartu, Institute of Computer Science, which gave me the opportunity to study the master degree in data science. Additionally, My sincere thanks also go to Prof. Dr Salwa Nassar from Electronics Research Institue, Egypt, who continuously believed in my strength with lots of support, encourage, and guidance.

I would further like to show my deep appreciation to my parents, and family for their love, unlimited help, and encouragement during all times of my life. Lastly, I dedicate my work to the memory of Prof. Dr Sherif Sakr, who continuously believed in my strength to be successful in the academic field.

1

Introduction

1.1 Introduction

Businesses and corporations are in need to develop their relationships with customers, and this happens by improving their business processes that need to be analyzed and assessed to get a significant improvement. Advanced technologies such as process mining and predictive process monitoring can play an essential role in evaluating, and analyzing business processes.

1.1.1 Process mining

A *business process* is defined as a combination of interrelated events, actions, and decisions that point at satisfying certain predefined results or enterprise objectives (2). A *business process* model is a conceptual design of a specific business process (3) and describes what input we received from the world and how software systems should react (4). *Process mining* goes beyond traditional dashboard for performance monitoring, and it is not focusing only on performance measures, e.g. numerical aggregates of performance, but it goes inside the box and opens it to dive into processes, activities, flows, and decisions to explain why the performance we are observing is the one that we are observing, and it connects the gap among regular process model outline and data-driven analysis, e.g. data mining, ML, and business intelligence (BI). At the core of any process mining technique (see figure 1.1), we have an *event log*, i.e. data we extracted from enterprise systems that contain information or transactions related to a given business process. e.g. *loan application*, and from there, we can do four types of operations. (i) we can discover a business process model from the event log *(discovery)*; (ii) If we have an event log and a process model we can compare them effectively and verify if the process model conforms to the event log or vice versa, e.g. the observed reality coincides with going to capture the process model *(conformance)*; (iii) Imagine if we have two event logs, e.g. the first one contains all cases when a customer was happy, and the second one contains all cases where a customer complained, and we need to understand what is the difference between the two logs *(variant analysis)*; (iv) Sometimes we have an existing process model with an event log, and we endeavour to describe the performance of the process on top of the process model *(performance mining)*. Interaction between world related systems, e.g. people, machines, and business processes that interact with software-related systems, e.g. bank and transportation systems leads to generate a huge amount of data that we call *event logs* that need to be analyzed and assessed.

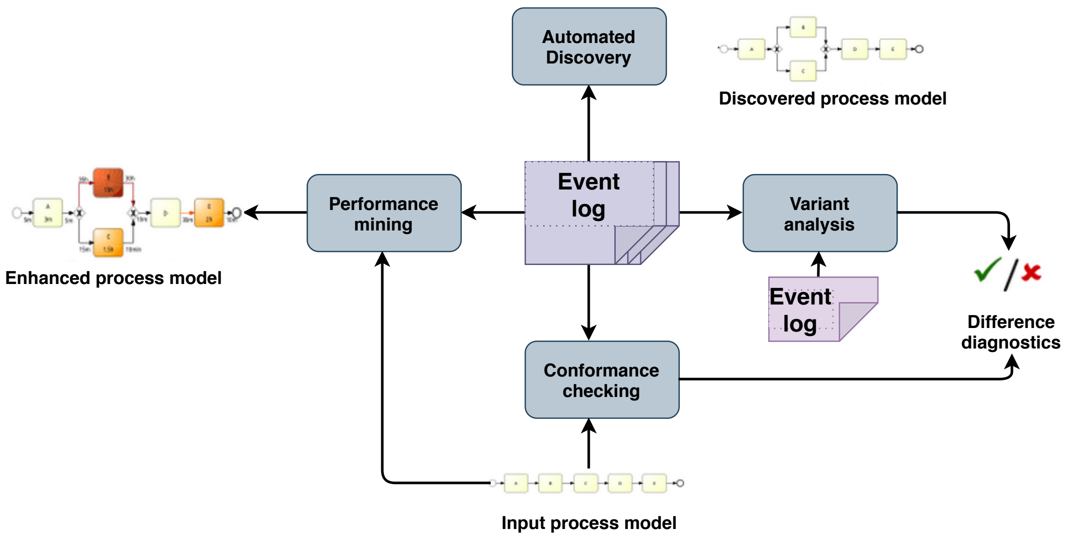


Figure 1.1: Process mining four main components (1)

An *event log* (see figure 1.2) is data we extracted from enterprise systems that contain information or transactions related to a given business process. e.g. *loan application* and its considered as a set of *cases (or traces)*, e.g. starting by submitting an application, making an offer until reach to an end state with acceptance or cancellation of the application. Each trace consists of a sequence of *events (or activities*), e.g. submit application, review document, etc., where each event in the trace belongs to the same case. Every event in the trace carries information about an *activity name* (describes what happened?), *timestamp* (indicates when an activity happened?), *resource* (who did the work?), and *life cycle* of an event (start and end of the activity). Event log attributes can be classified into two general groups (see figure 1.3) (5). (i) *Control flow*, and it contains the core attributes for any event log, e.g. *case id* (i.e. a unique identifier for a business process), *activity* (i.e. any action that can be done during the execution of business process), *timestamp* (i.e. start and end time for an activity); (ii) *Data payload* that refers to all other attributes of an event log, e.g. *name, age, etc.*

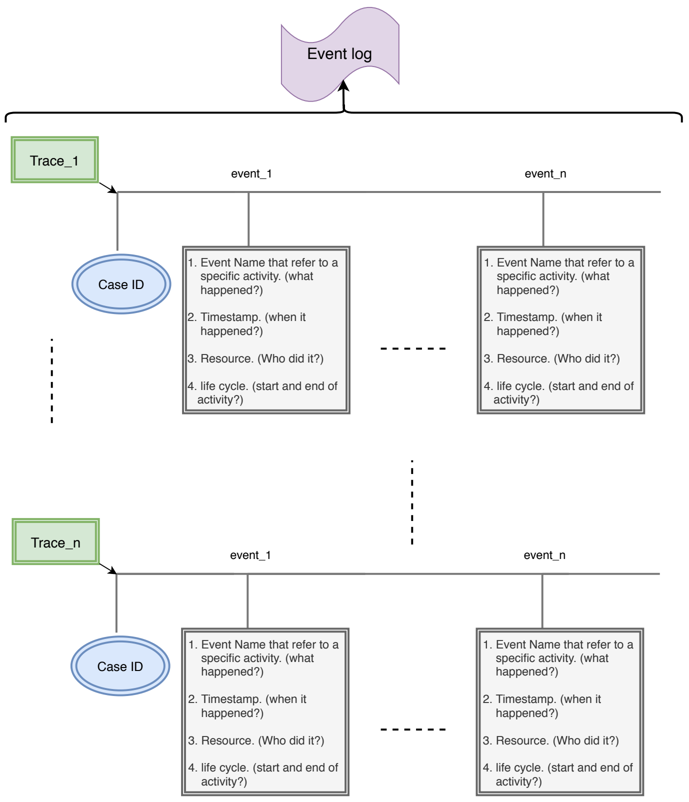

Event log
Trace_1
event_1
event_n
Case ID
1. Event Name that refer to a specific activity. (what happened?)
2. Timestamp. (when it happened?)
3. Resource. (Who did it?)
4. life cycle. (start and end of activity?)
Trace_n
event_1
event_n
Case ID

Figure 1.2: Event log: consists of a set of traces, and each trace contains information about its control flow and data payload.

Definition 1.1.1. ***Event*** Assume E holds the universe of all possible events, and

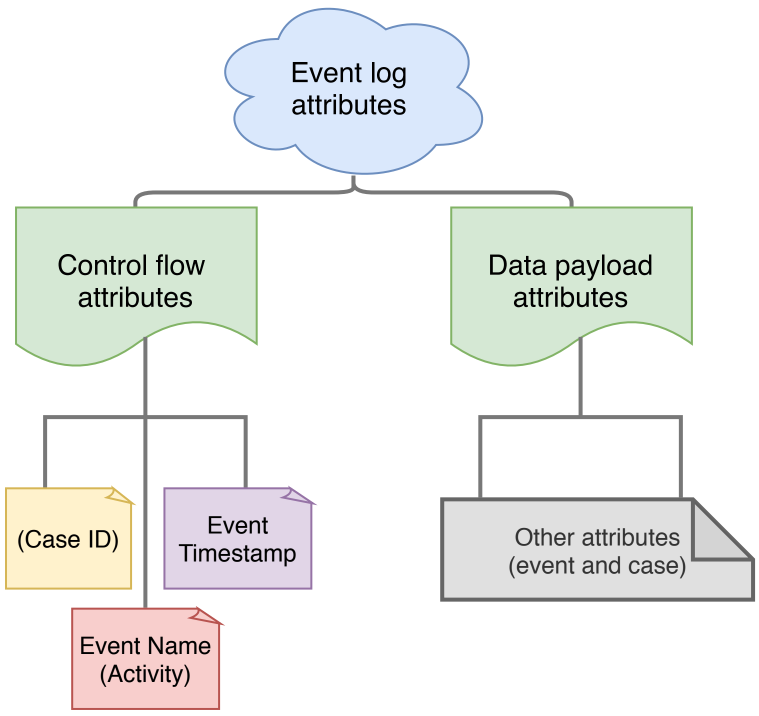


Figure 1.3: Event log attributes

$\forall e \in E : e = \{a, c, t, (m_1, ..., m_n)\}$ as a refers to an activity name; c is a unique identifier of a business case (or case id); t refers to the timestamp; finally, m represents all other attributes where $n > 0$

Definition 1.1.2. ***Trace*** Let R be the universe of all traces and $\forall \sigma \in R, \sigma$ is a sequence of events such that $\sigma = (e_1, ..., e_n)$, where all events in σ belong to the same case (see figure 1.2).

Definition 1.1.3. ***Event log*** An event log L is a collection of complete business cases (or traces) $L = \{\sigma_1, ..., \sigma_n\}$

Enterprises use process mining procedures with an event log as an input to answer a variety of key questions such as What is the process that people follow? And Where the bottlenecks in business processes? And Why? Where people deviate from the expected or idealized process? And Why? However, the process mining techniques presented above are focused on analyzing historical data and extracting insights about the general behaviour of the process. These techniques do not help us to monitor ongoing process instances nor to determine what will happen next? This latter question is addressed by a complementary set of process mining techniques known as *predictive process monitoring*.

1.1.2 Predictive process monitoring(PPM)

Predictive process monitoring (PPM) is a part of process mining and designed to help organizations and enterprises on a daily level. PPM takes a historical event log as an input (see figure 1.4) to build predictive models in the *offline* step then in the *online* phase we produce a stream of predictions based on a given stream of events about different ongoing business cases. In PPM given an *incomplete trace* (or running case) (see figure 1.5) different measures of interest can be predicted (6), e.g. outcome, next activity, or the remaining time for ongoing or running cases at their execution time. In this thesis, we are focusing on *outcome-oriented* PPM, which means we aim to predict the outcome for an incomplete trace. In this context, we define a prefix function as $P(\sigma, k)$ to return the prefix of a trace with size k.

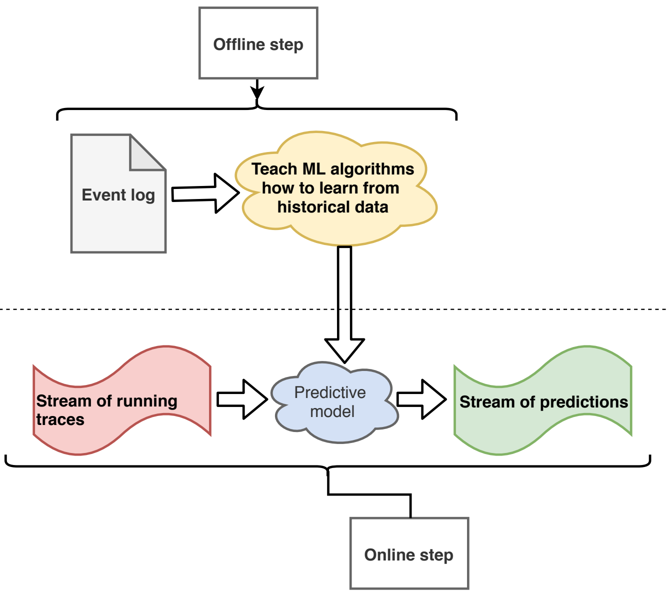


Figure 1.4: A general overview of predictive process monitoring workflow with its two main steps, i.e. offline (or training), and online (or prediction).

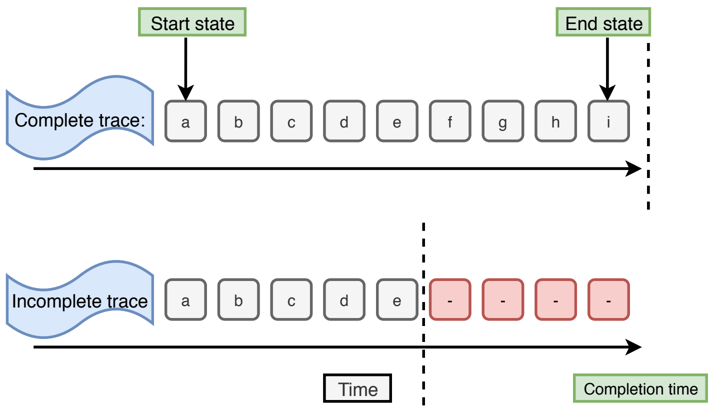


Figure 1.5: Difference between complete (i.e. it reaches to an end state) and incomplete (or running) trace (5).

Definition 1.1.4. ***Prefix function*** P Given a complete trace $\sigma = (e_1, .., e_n)$ where $k < n$, and $k \in \mathbb{Z}^+$, then $P(\sigma, k) = (e_1, ..., e_k)$

Organizations and enterprises have many different business goals *(outcomes)* based on their objectives, e.g. is a customer complaint or not. Accordingly, the outcome of a trace is determined using several methods based on business targets. An outcome can be *categorical* (i.e. discrete values, e.g. will the customer accept or decline a bank offer) or *numerical* (i.e. continuous values, e.g. invoice amount). In a business environment, its more suitable to define outcome classes as a binary variable with only two possible states that give information about if the ongoing case will end with an expected (i.e. ve^+, e.g. customer accepts the bank offer) or unexpected (i.e. ve^-, e.g. customer declines the offer) outcome. In this thesis, we deal with a categorical binary outcome for any business process monitoring task. To determine the outcome for a business case, we define a *target function* f.

Definition 1.1.5. ***Target function*** f A target function $f : R \rightarrow y$ maps all complete traces to its target class such that $\forall\sigma \in R$, then $f(\sigma) \in y$ where $y = \{1, 0\}$

Outcome-oriented PPM problems are usually solved by teaching (or training) a machine learning (ML) algorithm on how to learn from historical data, e.g. event log (see figure 1.4). A *predictive model* will result after the training phase, and then this model will be evaluated using test data (i.e. portion form historical event log). If the resulting model acts accurately on unseen (or test) data with acceptable accuracy, then it will be deployed to monitor the running cases in real-life which guides us to investigate how to improve existing PPM techniques in terms of performance.

1.1.3 Problem statement

This work answers the question of "How to improve the existing outcome-oriented predictive process monitoring methods?". The mentioned research question has been examined for a long time, which enriches the area of outcome-oriented predictive monitoring. Existing methods of predictive process monitoring have the same objective to improve business process performance. In this thesis, (i) we revisit most of the existing approaches for PPM (5); (ii) Introduce three methods to improve the outcome-oriented PPM; (iii) Create a benchmark of 20 different PPM tasks with the help of real-life event logs; (iv) Build a comparative experimental assessment of the existing and proposed methods utilizing this benchmark.

In this thesis, we combine and introduce three new approaches to improve the existing outcome-oriented PMM techniques.

1.1.4 Contributions

This work proposes three contributions to the area of outcome-oriented predictive process monitoring.

Contribution 1: Introducing a machine learning algorithm (i.e. *CatBoost*) (7) that handles categorical attributes automatically to outcome-oriented predictive process monitoring methods. *CatBoost* is a recently gradient boosting method and an open-source machine learning algorithm where its name has two main words that describe its features "**Cat**egory", and "**Boost**ing".

Contribution 2: Proposing a complex sequence encoding method based on *discrete wavelet transformation* (8) and time-series encoding. In this encoding process, we encode each trace in the event log into a collection of vectors with an equal length for each event in our log. In this contribution we used *neural networks auto-encoders* to reduce the dimensionality of the generated feature vector then we added this vector to other simple sequence encoding methods as a way to improve the overall performance of predictive monitoring in terms of accuracy.

Contribution 3: In real-life systems, the final outcome of an ongoing business process relies on the interaction between all business cases that are working concurrently at the same time (6). In this context, we propose new *inter-case* features that capture information about: (i) *Workload*, e.g. a number of cases that have started and did not finish; (ii) *Demand intensity*, e.g. how many cases were created since the current case started divided by a number of seconds since the running case was created; (iii) *Temporal context*, e.g. day of the week when the last event in the current case occurred.

This thesis is constructed as follows. In Chapter 2: we present the significant concepts and standards of machine learning to build a predictive model. Chapter 3: we summarize the literature for current outcome-oriented PPM methods. In Chapter 4: we propose our approaches to improve predictive process monitoring techniques. Chapter 5 we build the benchmark and conduct a comparative empirical evaluation. Lastly, Chapter 6 summarizes this thesis with a review of our proposed methods and addressing possible later improvement.

2

Background

In this Chapter, we clarify the vital principles of the machine learning field. We begin by showing various machine learning problems and available algorithms that we can use in each issue. After that, we give information about different evaluation metrics for each challenge. We continue with a brief description of classification algorithms that are utilized afterwards within the proposition. Finally, we end this Chapter with an overview of sequence encoding for classification problems.

2.1 Machine learning

People acquire new knowledge from prior experiences, and tools catch instructions produced by people, however, what if people can teach the tools to learn from historical data and do what humans can do and act much faster that's called *machine learning*—still, its a lot more than just learning; its also about understanding and reasoning. Machine learning is a group of techniques that learn from the past (historical) data on how to forecast or predict things related to the future. It tries to memorize or detect patterns from historical data that would help to predict trends in the near time.

Machine learning has a wide range of types and tasks, and we introduce the most important types of it in this thesis. For all types, we set up the following notation, (i) *Input:* $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ where $\mathbf{x}$ is a d-dimensional vector of independent variables or attributes, e.g. customer application; (ii) *Output:* y is the dependent (or target) variable e.g. good or bad customer; (iii) *Target function:* $f : X \to Y$ e.g. ideal credit approval formula; (iv) *Data:* $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ e.g. historical records; (v) *Hypothesis* $g : X \to Y$.

ML can be classified into three main classes: (i) *Supervised learning:* (x, y) in the form of *(input, correct output)* where historical data as input, e.g. event logs are available with its correct label, e.g. outcome. This type of learning can be split into two main problems: Firstly, *classification* where ML algorithms (e.g. *Random Forest, XGBoost, CatBoost,* etc.) learn how to predict *categorical* values such as good or bad; Secondly, *regression* where ML techniques (e.g. *Linear regression, Ridge Regression,* etc.) predict a numerical and continuous value such as a price of the house; (ii) *Unsupervised learning:* $(x, ?)$, in the form of *(input, ?)* where historical data as input, e.g. users and movies characteristics are given with no information about the target label at all. In this type, we dont predict a particular label regarding the input data. Still, we learn how to extract hidden structures from the give data (e.g. *k-means*, i.e. Clustering algorithm), e.g. group customers to high/med/low level; (iii) *Reinforcement learning* which is an intriguing type of learning because it represents the human's experience in learning, e.g. toddler learn to not touch a cup of tea that have smoke coming out of it. It's mostly used in games where an agent is in a place where it should take a set of *actions*, and based on the action; the agent will be *rewarded.* It takes the form of $(x, \hat{y} \subset y$, grade), i.e. *(input, some output, a grade for this output).* Most machine learning methods assume that all attributes are *numeric.* For *categorical* ones we convert it to a numeric using different methods such as *one-hot* encoding, however, in this thesis, we dont need to do so since we introduce a machine learning method that can handle categorical features. Most of the ML methods mentioned above are using the same workflow across different projects; however, in this thesis, we interested in supervised ML methods as outcome-oriented PPM task is considered as a classification problem.

2.2 Machine learning workflow.

In this section, we introduce the general and basic workflow for any supervised ML project (see figure 2.1), including the work on this thesis. (i) At the beginning of an ML project, data preprocessing must be done, and mostly it takes most of the

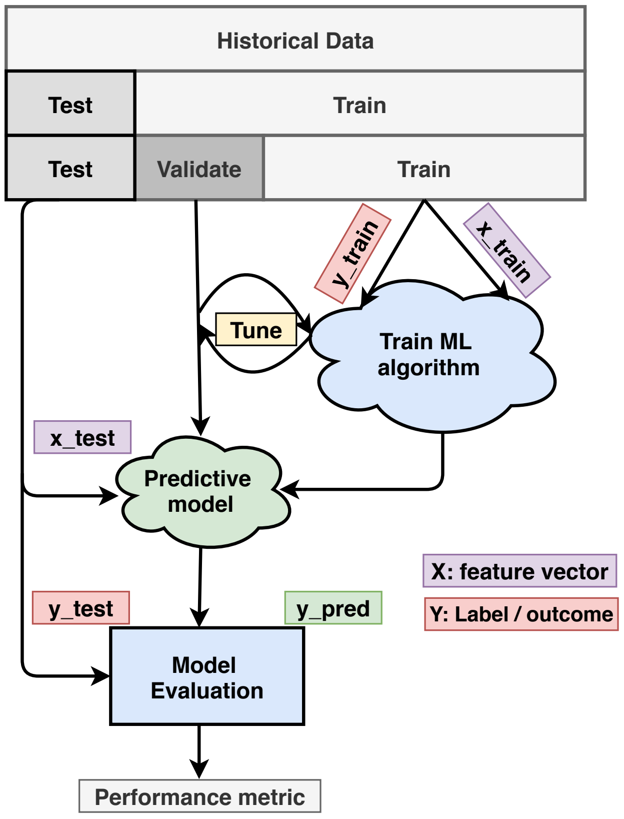


Figure 2.1: Machine learning workflow: split data into train, and validate sets to train and tune a machine learning model, in addition to a test set to evaluate this model.

time, however in this thesis we used prepared data from (5) which represents 20 real-life event logs from around 7 different disciplines. (ii) Then we move forward with building a predictive model using different classification algorithms such as CatBoost. (iii) Splitting the historical data, e.g. event logs into *training* (i.e. used to learn pattern form event log) and *testing* (i.e. used to understand how the predictive model would perform in real-world cases) data sets. (iv) Tuning by splitting the training data further into a *validation* (i.e. used to understand the predictive model behaviour on unseen data by calculating loss error then the model will adjust its parameter based on evaluation results on it) and training sets, this step is very important during the learning process, however in this thesis, we used *hyperopt* (9) (i.e. a Python package to tune *hyper-parameters* (i.e. variables we give to the algorithm during the learning process and cant be learned, e.g. learning rate) of ML algorithm automatically) to do this step. Additionally, it's important to note that in traditional machine learning, the splitting of a dataset into a train, test, and validate sets is done randomly; In other words, a random subset of objects are put in the test set, and the rest are randomly split into a validate and train sets. However, in the field of PPM, it is important to consider the fact that the input data is temporally arranged, and that, naturally, one cannot use data that is only available in the future (relative to a given point in time) in order to make predictions about the future. Accordingly, in this thesis, we use a *temporal approach* to split between train and test data, as proposed in (5). Specifically, the traces in the event log are ordered according to the start time of their first event. The first traces in this chronological order are put into the training set, and the remaining into the test set. The traces in the training set are then further split into train-validate using the same principle. (v) The last and final step before deploying and delivering the predictive model to business organizations is the *evaluation* of the resulted model after the training process.

2.3 Evaluation metrics

After building a predictive model, we evaluate it to decide if the resulting model is good enough to make predictions or not. To quantify the goodness or badness of classification models, we measure different evaluation metrics on test (or unseen) data by comparing (see figure 2.1) the predicted values (i.e. *y_pred*) from the model with its true label (or baseline, i.e. *y_test*). In the following subsection, we introduce different evaluation metrics. From now on, we discuss a binary classification problem i.e. $y \in \{1, 0\}$ or $y \in \{+^{ve}, -^{ve}\}$.

A starting spot to quantify the goodness and badness of a classification model in ML could be a *confusion matrix* (see figure 2.2). It shows the relation between correctly classified cases (or samples) and incorrectly classified cases. Confusion matrix split into four groups: (i) *True positives (TP)* which indicate to the actual true outcome (i.e. *y_test*) was positive (i.e. 1), and its predicted correctly (i.e. *y_pred*, e.g. 1) to be positive by the model; (ii) *False positives (FP)* (i.e. *type 1 Error*) refers to the actual true outcome was negative (e.g. 0), and its assigned by the predictive model as positive; (iii) *True negative (TN)* it means that actual true outcome was negative and the model predicts it correctly to be negative; (iv) *False negative (FN)* (i.e. *type 2 Error*) refers to the actual true outcome was positive, and it predicted incorrectly to be negative by the model. Using confusion matrix we get different *metrics* to evaluate classification models such as *accuracy*, i.e. relation between correctly classified cases and the total number of cases that are classified correctly and incorrectly, *recall*, i.e. it gives information about how many models predict cases correctly out of all positive cases, *precision*, i.e. how many cases that are positives, and model predict them successfully, and *F-score* gives us the ability to measure precision and recall at the same time, and it uses *harmonic mean* between them since its challenging to compare high recall with low precision together or vice versa.

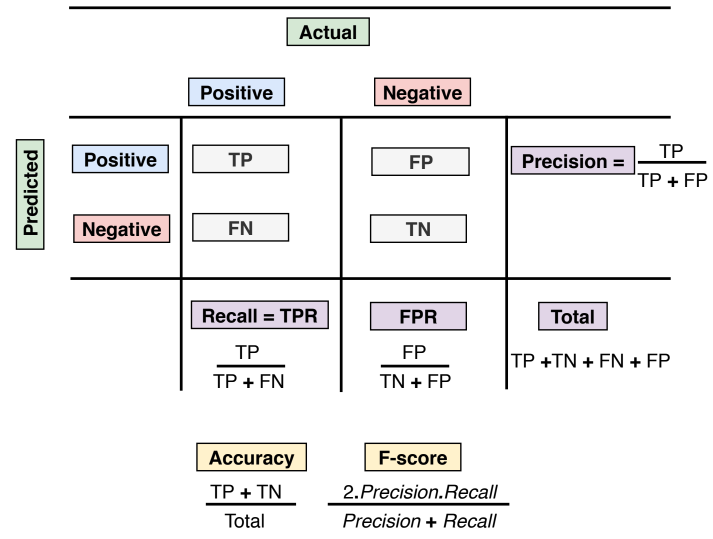


Figure 2.2: Confusion matrix: represents the relation between actual outcomes and predicted cases. It contains different metrics that we use to evaluate any classification model, such as accuracy, F-score, etc.

The Most used metric in classification problems such as outcome-oriented PPM tasks is *accuracy*. However, in many other cases, its not suitable to use it, such as when we have imbalanced outcomes where the majority of cases are negative in comparison to positive cases. In this situation, its recommended to use an *F-score* that gives information about precision and recall at the same time. All these metrics consider that the model will predict a binary number, i.e. one or zero based on the positivity or negativity of the predictions, however in many other cases the output from classifiers could be a probability estimation of the predicted outcomes and based on a threshold ($\tau = 0.5$) we can assign each estimated value to a specific class, e.g. model predictions are 0.2 and 0.8 and based upon that we assigned 0.2 to be negative and 0.8 to be positive predictions.

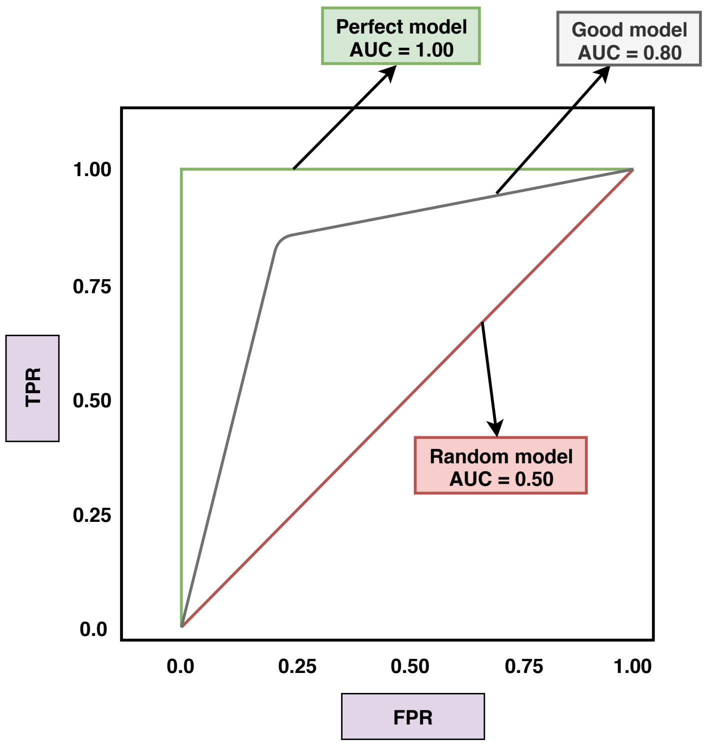


Figure 2.3: Receiver operating characteristics (ROC) curve show the relationship between the true positive rate (FPR) and the false-positive rate (FPR). The highest AUC the more robust predictive model.

Two more things that we can get from confusion matrix are (v) *True positive rate (TPR)* that is the same as recall and sometimes called *sensitivity*; (vi) *False positive rate (FPR)* that is opposite to recall, i.e. it gives information about how much a model predicts cases correctly out of all negative cases. We used in this thesis something else which is a widespread technique to evaluate classifiers that output a binary value score by building a *Receiver Operating Characteristics (ROC)* curve by plotting (see figure

2.3) pairs of TPR and FPR. Given a threshold AUC, i.e. the area under the ROC curve (or area under the curve) is mostly used to represent the relation between TPR and FPR as a single measured value. Example of the various advantages of using AUC as a single measure of outcome-oriented PPM over accuracy and F-score is that it keeps unbiased with imbalanced cases (10). We can use all the mentioned metrics above in classification problems; however, in ML, there are several classification algorithms that can be used to solve different tasks. In this thesis, we deal with outcome-oriented PPM task, and we used five different classification algorithms to tackle this problem.

2.4 Classification algorithms

In supervised ML, we have two main types of problems, as mentioned earlier: classification and regression. In this thesis, we deal with outcome-oriented PPM, which is counted as a classification problem. In this part, we give an outline of different classification algorithms that are used in this thesis. We propose a recent classification algorithm to the area of outcome-oriented PPM, i.e. CatBoost with other four (e.g. XGBoost, Support Vector Machine, Random Forest, and Logistic Regression) classification algorithms from (5).

1. *Decision trees:* DT is a type of supervised ml algorithms that is mostly used in classification tasks, but it can be used in regression tasks as well. DT is known as *CART*, i.e. classification and regression tree. A DT can be constructed as a flow chart since it has the same structure as it. The start and first node in a tree is the root node where each inner node indicates a test on each independent variable, and each division shows the outcome of that test, finally, the end node of the tree (or leaf node) carries a specific label, e.g. positive or negative. The advantages of CART are simple to understand, interpret, and visualize. DT performs a feature selection implicitly, and it can handle both numerical and categorical variables. The disadvantages of DT is *over-fitting* (happens when the model learns data more than enough). DT can become uncertain because little changes of the data strength result in totally diverse produced trees, i.e. modification that requires to be lower by *ensemble approaches (e.g. bagging and boosting).* To construct a decision tree we want to decide about four main things: (i) Which features to choose to start splitting; *Information gain* is one of the methods that are used

to choose between independent variables, and our target is to start splitting with features that have maximum information gain, i.e. capture more information. (ii) What condition for splitting; (iii) When to stop splitting; (iv) How to prune a tree to reduce the possibility of over-fitting; Trees are very useful when we use it with another superior machine learning techniques, e.g. Random Forest and Boosting (will be discussed later). DT has many parameters that we can tune such *max_depth* that defines the maximum size of a tree or how much deeper we can go in the tree which means with a large max depth we can do more splits and capture more information about the data.

2. *Random Forest:* RF is a supervised ML algorithm worked with both classification and regression tasks and acts as a massive number of uncorrelated decision trees. RF uses *bagging* (i.e. the mixture of learning models improves the classification efficiency) concept, and the principal purpose of bagging is to average boisterous and unbiased models to build a model including few variances. As the name implies, this method generates a forest with a number of decision trees, and in common the larger trees in the forest, the higher robust the prediction and moreover better accuracy. To model multiple decision trees to create the forest, we use the same methods of constructing the decision with the information gain as the DT algorithm. In RF, we develop many trees as opposed to individual tree created in the CART method. To classify a distinct business process case based on all independent variables, every tree votes for a specific outcome class then the forest chooses the classification with the largest votes across all other trees in the forest. The advantages of RF classifiers are that they handle blowing values and preserve accuracy when a huge amount of the data are missing, handle large data sets with high dimensionality, and will not over-fit the model compared to DT. RF parameters that we can tune are the same as in decision trees, e.g. *max_depth*, in addition to the *number of iteration*, which represents the number of trees in the forest.

3. *Gradient Boosting:* GB is a type of machine learning algorithms that work well with heterogeneous data, i.e. does not have a specific internal structure and for this type of data gradient boosting is giving the best solution. In addition to the simplicity of using it. It is an iterative algorithm that is based on decision trees.

GB is a group of ML procedures that contain multiple different algorithms such as *XGBoost* (extreme gradient boosting), and *CatBoost* (categorical boosting). It is estimated one of the common and most influential machine learning methods that can provide great results in terms of accuracy in many different practical tasks and can solve many complex data-driven real-life problems such as outcome-oriented PPM. These methods depend on the *ensemble* (i.e. create many weak models to work together to create a very strong model) learning, and there are two main popular ensemble methods: (i) *Bagging* (see RF); (ii) *Boosting*; In Boosting, predictive models are implemented in series, and in every progressive model the weights or the parameters are customized w.r.t the learning of the earlier model. GB is a special kind of boosting methods where it operates on decreasing faults sequentially and by tuning its parameters such as *learning rate* (control the speed of learning process for each tree), *max_depth*, *number of iteration*, etc., we get very good results in terms of accuracy.

4. *Logistic Regression:* LR is a learning algorithm that we use when the outcome of the case is either positive or negative, i.e. binary value. This learning algorithm is considered as one of the simplest techniques for classification problems given set of independent variables $x = (x_1, \ldots, x_m)$, the LR algorithm will calculate $\hat{y} = P(y = 1 \mid x)$ by finding a linear combination of all independent variables (features) by giving each variable weight and sum the weighted sum of all features. $\forall x \sum_{i=1}^{m} w_i \cdot x_i + \epsilon$, where w_i are model *coefficients*, and it's considered as a learnable parameter that LR algorithm can learn during the training process and ϵ is the *intercept.* After that, the weighted sum is given to a sigmoid function σ to convert it to an estimated probability value for both outcome classes such as $\sigma(\sum_{i=1}^{m} w_i \cdot x_i + \epsilon)$. Fitted parameters such as coefficient, and intercept can be learned through optimizing a *loss function*, e.g. *cross-entropy loss* that measures how good our model is in only one training sample (or case), and to generalize this loss to all training cases we calculate *cost function* which is the average losses of the entire training set. To learn these parameters automatically, we use a *gradient descent optimization* algorithm to iterate over all coefficients (or weights) and update them during the training process.

5. *Support Vector Machine:* SVM is a learning method which acts similarly to logistic regression when it comes to linear classification problems, in addition to the ability to do nonlinear classification through the *kernel trick* that maps the data to a high dimensional feature vector. SVM works by discovering a *hyperplane* that can separate the two classes. There are many hyperplanes that can be discovered using SVM, but the idea here is finding a plane with *maximum margin*, i.e. maximum distance between cases for positive and negative class and *hinge loss* is used to help with optimizing the maximum margin of SVM. In SVM we can optimize three different parameters to get better results in terms of accuracy: (i) *Kernel:* converting the input from a low dimensional space into a high dimensional space (e.g. *Radial Basis Function (RBF), sigmoid, polynomial, etc.)*; (ii) *Regularization (C):* this parameter used to represent the error of misclassified cases, and its used with the decision boundary to control the trade-off between them; (iii) *Gamma:* this parameter can be set with RBF kernel, and it refers to how far/close influence a single training case has (11).

2.5 Early classification and sequence encoding approaches

About the extensive writing on ML, we found that the outcome-oriented PPM belongs to the task of an early sequence classification problem. It means, given a collection of sequences with its outcome (or label), the target is to define a function F that for each sequence prefix it predicts the outcome of w.r.t this prefix.

In recent years, early sequence classification approaches are by and large centred on identifying a *prefix length* that gives a reasonable prediction (12). In literature, methods to define the range of prefixes, are many and different. For example (13), they built a way that helps with early sequence prediction problems that depend on the relation between accuracy of the prediction for the prefix and accuracy for the full sequence.

Outcome-oriented PPM is considered as a problem of an early sequence classification with *complex sequence encoding* (i.e. a stream of events with payload attributes) of the event logs (14, 15). Complex sequence encoding methods are proposed in (16) where it used in place of *simple sequence encoding* (i.e. a stream of events without payload) to deal with outcome-oriented prediction problems. However, the complexity space of the prediction problem increases significantly (17). A few works have been done on early sequence classification over complex sequence encoding, for example (18), where they develop a serial decision tree to determine the length of prefixes. A few of the baselines of dealing with complex sequence encoding utilize *index-based* encodings (16). Complex sequence encoding permits us to encode an event log, to capture as much data as conceivable, whereas having a structure that effectively can be utilized with standard machine learning algorithms.

3

Literature review

In this Chapter, we provide a taxonomy of the existing outcome-oriented PPM methods for business manners, including an overview of current techniques for sequence encoding, trace bucketing, and evaluation measures. In recent years the area of PPM becomes very rich and has many different principles and evaluation methods, so we decided to focus on the outcome-oriented PPM only.

This Chapter summarizes state of the art in the field of outcome-oriented PPM based largely on an existing literature review (5). In section 3.1, we present existing techniques for building and training a predictive model using a historical labelled data of an event log that contains a set of complete traces with its outcome that can accurately predict incomplete business cases during its execution time. Wherein section 3.2, we introduce different existing approaches to quantify the goodness or badness for PPM methods.

3.1 Outcome-oriented PPM methods and Taxonomy

In this section, we include a taxonomy of the existing outcome-oriented PPM methods for business processes. Our aim from this section is to answer questions about What are the current techniques to train outcome-oriented predictive models? And How to combine these methods? Within the taking after subsections, we begin with presenting the concepts and the workflow for outcome-oriented PPM. We continue with characterizing the existing approaches concerning the diverse steps discernible in the workflow and conclude with a scientific categorization for the examined methods.

3.1.1 Outcome-oriented PPM Basics and workflow.

All reviewed methods use universal principles and concepts. We defined some of them in Chapter 1 (e.g. event log, trace, etc.) In outcome-oriented PPM, forecasts are formed using a supervised machine learning method (or classifier) which uses as input data a set of independent attributes X (or feature vector) and learns a function f that maps $X \rightarrow y$ where y is the outcome of the ongoing business case. To feed a classification model with an event log as input, we define *trace encoder* to encode all traces in this log into a feature vector.

Definition 3.1.1. ***Trace encoder*** A trace (or sequence) encoder *seq_enc:* $R_* \rightarrow X_1 \times \cdots \times X_d$ is a function that takes a business case σ and encodes it to a numeric feature vector toward the d-dimensional vector space $X_1 \times \cdots \times X_d$ with $X_k \subseteq \mathbb{R}, 1 \leq k \leq d$ being the domain in k-th independent variable.

Definition 3.1.2. ***Classifier*** A binary classifier *clf*: $X_1 \times \cdots \times X_d \rightarrow [0, 1]$ is a method f that takes d-dimensional feature vectors as input and maps each vector into a target label through a prediction score reflecting the probability of the positive target.

In predictive process monitoring, we teach a clf on how to learn from a historical *prefix log* during the training operation.

Definition 3.1.3. ***Prefix log*** A prefix log $E_* \subset E$ is an event log that includes a subset of prefixes of E. The prefix log contains all possible prefixes from an event log. Prefixes can be grouped into buckets using a bucketing function and for each bucket, B we train a separate classifier clf.

Definition 3.1.4. ***Bucketing function*** A bucketing function $bucketer : R_* \rightarrow \mathbb{N}$ is a method that takes a business case σ and put it inside a bucket b_i where $1 \leq i \leq B$, and B is the total number of all possible buckets.

In this context and taking into account these concepts, we describe a predictive model as follows.

Definition 3.1.5. ***Predictive model*** A predictive model is a general function $Pred : R_* \rightarrow [0, 1]$ that takes a running case (or trace) σ as an input and do three main things based on the above definitions: (i) assign this trace into a particular bucket b_i based on the bucketing function; (ii) encode σ into a feature vector based on the *seq_enc* function; (iii) estimate the probability of the positive outcome based on $Pred$ function. In other words $pred(\sigma) = clf_{bucketer(\sigma)}(seq\_enc(\sigma))$.

The common workflow for outcome-oriented PPM can be viewed as two main steps (5). (i) *Offline* step to teach (or train) a predictive model on how to learn from historical event logs (see figure 3.1), this step can be decomposed into four steps. First, we extract all possible prefixes from a given event log. Next, Based on the similarities between the extracted prefixes, we separate them into a set of buckets. After that, we encode each bucket into a feature vector to train a separate classifier for each bucket which is the final step. (ii) *Online* step, where we can predict the final outcome for the running traces using the trained classifiers, and prefix buckets form the offline step. Particularly, for a running case, we determine the corresponding bucket from historical buckets first, then we encode this running trace into a feature vector using an encoder, and finally based on the selected bucket we chose the corresponding classifier to predict the final outcome of this case. In the subsequent subsections, we explain the four steps of the offline step and utilize this description to build a taxonomy about existing methods for outcome-oriented predictive process monitoring.

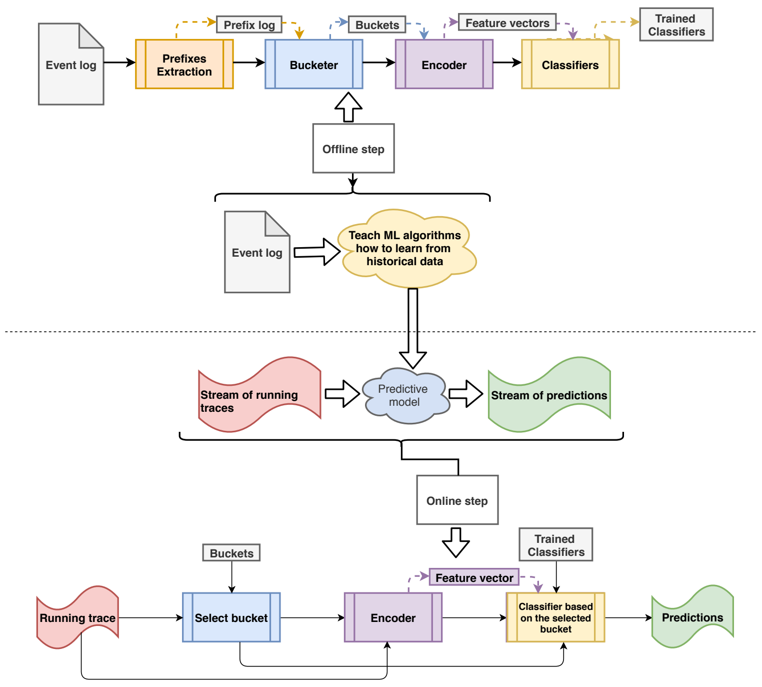


Figure 3.1: A detailed predictive process monitoring workflow, including trace bucketing, sequence encoding, and classification for online and offline steps.

3.1.2 Prefix extraction

In outcome-oriented PPM, we aim to predict the final outcome of the ongoing business case (running trace), it means the unseen data for our classifier is a prefix log. Because of that, its obvious to train this classifier on a prefix log as well, to make training and testing operations comparable. In this context, we define how to extract prefixes from a historical event log.

In this thesis, we are dealing with 20 different event logs from real-life and extracting all possible prefixes from these logs may raise many problems during the training process for predictive models. (i) Different length of the business cases in the original event generates different prefixes; in other words, more prolonged cases generate more prefixes and affect the training process by making classifiers to be biased to them. (ii) The training process with a large number of prefixes is a bottleneck because it will take more and more time and slow down the whole training process. Thereby, it is standard to extract prefixes from the original event log up to a specific number. In (16), they extracted prefixes up to 30 events only, at the same time (19) limits the length of prefixes up to 20. Based on the similarities between the extracted prefixes, we separate them into a set of buckets using a trace bucketing technique.

3.1.3 Trace bucketing

In outcome-oriented PPM, training multiple classifiers for different buckets is very common. Based on the similarities between prefix traces in the original event log, we can divide them into different buckets, then train a binary classifier for each bucket individually. During the execution time of any ongoing case, we select the most suitable bucket for that case, after that apply the respective classifier on it to predict the outcome of that case. In this subsection, we present the bucketing methods that have been introduced by current PPM techniques.

Single bucket. This approach has been introduced in (20). In this method, all prefix traces are not split into different buckets and considered to be within the same bucket (single bucket). Accordingly, there is no need to train multiple classifiers in these methods, after that we apply the trained classifier to the ongoing cases during their execution time.

KNN bucketing. This method has been introduced in (21). This method does not follow the same general workflow we introduced in section 3.1.1 and skip the offline step, in addition to the prefixes are not split into different buckets. Alternatively, for any prefix trace and based on the KNN algorithm that selects K nearest neighbours for any training or testing sample, it selects the k nearest neighbours (i.e. based on similarities between traces using string-edit distance) for that trace then puts them into a single bucket and trains a classifier on top of the generated bucket. In this context, the number of buckets and classifiers are not fixed and grow based on incoming events.

Cluster bucketing. This approach was introduced by (22). In this method, the trace prefixes are split into different buckets (or clusters) using various clustering algorithms such as *DBScan*, or *hierarchical clustering*. Accordingly, we train a classifier for each resulting cluster by taking into account prefix traces on each cluster only. During the execution time of the ongoing case, a cluster is selected based on historical clusters; then the relevant classifier is applied. In (22), they used two different clustering algorithms, the first one depends on *string-edit* distance, i.e. *DBScan*, and the other one based on *Euclidean distance*.

State bucketing. In this approach prefix traces are divided into buckets based on the similarities between trace states, i.e. each component (e.g. task) in the process model is relative to a particular *state*). During the execution time of the ongoing case, a bucket is selected based on historical buckets; then the relevant classifier is applied. This approach depends mainly on having the process model earlier. In the literature, there are predictive methods based on the state approach that take a process model and an event log as an input, and they assume that the process model meets the event log accurately such as (23, 24, 25). Alternatively, in (26), they introduced a method to directly generate a process model of the given event log.

Prefix length bucketing. This method has been introduced in (16, 27). The idea of this bucketing method is to build a bucket for each prefix length. For example, the first bucket contains three events from a trace, and the second bucket includes four events from a trace, and so on. For each bucket, one classifier is trained to predict the ongoing cases during their execution time.

After splitting all traces into different buckets based on the above techniques, we encode each bucket into a feature vector using different sequence encoding methods.

3.1.4 Sequence encoding

To teach a machine learning algorithm how to learn from a historical event log, all prefixes in all buckets need to be *encoded* into a feature vector. This encoding step is a little bit challenging because of the nature of event log attributes. *Event attributes* are *dynamic* and change during all incoming events; therefore, more data regarding the case becomes accessible. On the other hand, *case variables* are *static* for the whole business case. To tackle this problem a trace abstraction technique (28) can be used; after that, we use feature extraction methods to encode prefixes into feature vectors. However, in PPM area, we need to distinguish between two types of features that we can encode into a feature vector. (i) *intra-case*: which we can extract from the running business case or in other words, information that we can get from only one business case. (ii) *inter-case*: that we can extract from the concurrent execution of all business cases or information that we obtain from all business cases together. Most of the existing sequence encoding techniques such as *last-state, aggregation, and index encoding* depend only on intra-case features. However, several previous research studies, such as (6) have investigated the question of how to supplement intra-case with inter-case features. These studies have highlighted that inter-case features can bring additional information into the classifier, and accordingly, better performance can be achieved in terms of accuracy.

In this context, many sequence encoding methods have been surveyed in (16), and (5). In the coming paragraphs, we present different encoding methods that can be used in outcome-oriented PPM problems.

Static encoding. This method was introduced in (16). Case attributes remain the same within the whole case that makes the encoding process very simple, and we keep it without any loss of information. To exemplify categorical attributes as numeric features *one-hot* encoding or *get dummies* methods can be used.

Last state. In this approach, only the very last m event attributes are measured. For this reason, the feature vector size is fixed during the execution of a case and proportional to the number of event attributes. A drawback of this approach is that anything that happened in the past and capture information will be neglected, i.e. information loss. This encoding method is commonly used with the KNN method (21), and state-based bucketing (24).

Aggregation. In this approach, the drawback in the last state method that is *lossy encoding* is solved by considering all events since the beginning of the business case. Many aggregation functions are used with numerical attributes such as *max, min, average,* and *sum* as well as with categorical attributes such as *count* (20) to make sure the size of the feature vector is constant. Although this method considers all events, it neglects the order of events which is the drawback of this technique.

Index-based encoding. This method was introduced in (27). It uses all possible information in the case by considering all events as the aggregation method and also the order of events. The profit of this procedure is that it is considered a *lossless* (i.e. we can construct the original trace from its feature vector) encoding method. A drawback of this method is it can be used only with homogeneous (i.e. all traces have the same length) buckets.

All encoding methods are dealing with event attributes except the static method where it deals with case attributes. Figure 3.2 shows a summary of all encoding methods with trace abstraction. After encoding each bucket into a feature vector using the above techniques, we use a classification algorithm to train it on those vectors to build a predictive model.

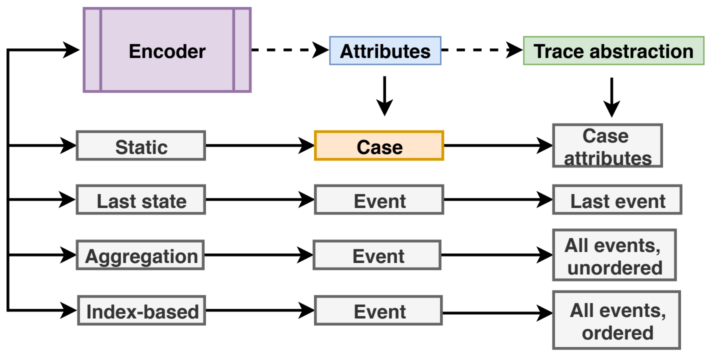


Figure 3.2: Taxonomy of current sequence encoding techniques for PPM such as static, last state, etc., and trace abstraction for each technique.

3.1.5 Classification algorithms

Outcome-oriented PPM task is considered as a classification problem that aims to predict the outcome on ongoing cases during its execution time. Current outcome predictive methods have been examined with many classification algorithms. Decision tree algorithm is The most popular classifier that is used for PPM tasks (20, 21, 23). Also, RF has been used in many studies before, such as (16, 29). Furthermore, SVM (25), KNN (27), and Gradient Boosting methods (5).

3.1.6 Discussion

Outcome-oriented PPM is a very growing research area. In this Chapter, we gave a complete analysis of existing outcome-oriented PPM methods that aim to predict the outcome of a business case during its execution time. On the basis of the above, we conclude that prefix extraction methods are not characteristic to any given PPM techniques. Alternatively, these methods are chosen on the basis of the performance measure and can be utilized by all PPM techniques. In the same context, machine learning algorithms are growing very fast, and there is no rule of thumb to use a particular algorithm with particular business process data, and again the performance metric is the judge to choose between different algorithms. To categorize current outcome-oriented PPM, we should identify two main things: (i) How to split prefix traces into several buckets; (ii) How to encode each prefix in each bucket into a feature vector. To answer those two questions, we provide a taxonomy (see figure 3.3) of the appropriate methods based on those two viewpoints plus different classification algorithms that have been used for outcome-oriented PPM. A predictive model can be built using different sequence encoding, trace bucketing, and classification methods. However, before deploying this model in a real-life environment, it should be evaluated to measure the power of it in real business cases.

3.2 Evaluation measures

To quantify the goodness or badness of an outcome-oriented PPM, different evaluation measures can be used. In this section, we provide possible evaluation measures in the literature. In outcome-oriented PPM a right prediction ought to be correct (or

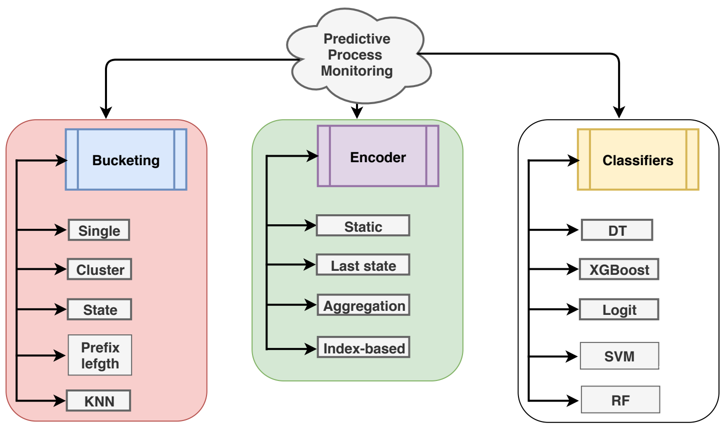


Figure 3.3: Taxonomy of existing sequence encoding, trace bucketing, and classification methods for outcome-oriented PPM.

accurate) and it ought to be executed in the beginning steps of the operation as well as it should be produced efficiently. In the following passages, we dive into these measures.

Accuracy. It refers to the relation between the correctly classified cases compared to the actual true labels, i.e. ground truth and all classified cases. Inaccurate predictions are useless as it does not help with making decisions. Accordingly, accuracy is considered as the most important measure of a prediction. Classification models give output as a real-valued score to estimate how likely this case belongs to a positive or negative class or binary values that reflect a positive or negative class. In binary values outcome, it is common to use metrics that we can derive from the confusion matrix (see Chapter 2), e.g. accuracy, precision, recall, and F-score. In real-valued outcome, the most commonly used metric is AUC, i.e. area under the curve. In continuous predictive monitoring (i.e. the system worker gets predictions about the ongoing cases after each event where the system doesn't tell the worker how to intervene to avoid undesired outcome) prediction accuracy is mostly stated at different prefix lengths or evaluation points (16, 19, 27). Furthermore, the net accuracy is calculated as the average of accuracy through all evaluation points.

Earliness. In predictive process monitoring, it is imperative to predict the final outcome as soon as possible to give the process worker the ability to act against unexpected outcomes. In continuous predictive monitoring, earliness is measured at different evaluation points implicitly or explicitly (16).

Efficiency. In PPM systems, it is important to predict the outcome in an acceptable time. Accordingly, Execution time taken during online and offline steps are computed to report the efficiency of a model. In (22), they split the execution time taken during the online step into *processing time*, i.e. time was taken to encode and assign test prefixes into buckets and *prediction time*, i.e. the average time to predict the outcome for a given prefix.

3.3 Summary

In this Chapter, we summarized the state of the art in the field of outcome-oriented PPM based largely on an existing literature review (5). At the same time, we give a review of the current outcome-oriented PPM approaches, including sequence encoding, trace bucketing, and classification methods. In the next Chapter, we start by introducing the methods we propose in this thesis to improve the outcome-oriented PPM techniques.

4

Proposed Methods

In Chapter 3, we recognized three essential gaps in the related work. Firstly, the potential of using new machine learning algorithms that handle categorical variables, specifically *CatBoost*, to predict the outcome in the context of business processes has not been explored. Then, Exploring different sequence encoding methods to obtain a lossless encoding of the feature vector, in particular, *Discrete Wavelet Transformation.* Finally, studying the potential of using *inter-case* features since in real-life systems the outcome does not depend on the running case only, but it depends on features that come from the concurrent execution of all business cases.

This Chapter will clarify the subsequent research question:

RQ1 How to improve the current outcome-oriented predictive process monitoring approaches in terms of performance?

To answer RQ1, we propose solutions to the mentioned three gaps, starting with CatBoost in section 4.1, followed by wavelet transformation in section 4.2, and ending with inter-case features in section 4.3.

4.1 CatBoost

CatBoost is one of the most potent and recent gradient boosting (i.e. a type of machine learning algorithms that work very well with heterogeneous data, i.e. does not have a specific internal structure, that is based on decision trees) algorithms. Motivation to use CatBoost in outcome-oriented PPM comes from the great results that are shown on a variety of publicly available datasets. CatBoost is very stable to parameter changes (7) which means using it with default parameters often wins all other methods with tuned parameters.

Catboost (i.e. Categorical Boosting) differs from other gradient methods through three main things: (i) Way to build trees; (ii) Categorical features handling; (iii) Boosting method; We will compare CatBoost algorithm with other gradient methods (see figure 4.1) such as XGBoost (i.e. Extreme Gradient Boosting), and LGBM (i.e. Light Gradient Boosting Machine) to show why we introduce this method into outcome-oriented predictive process monitoring.

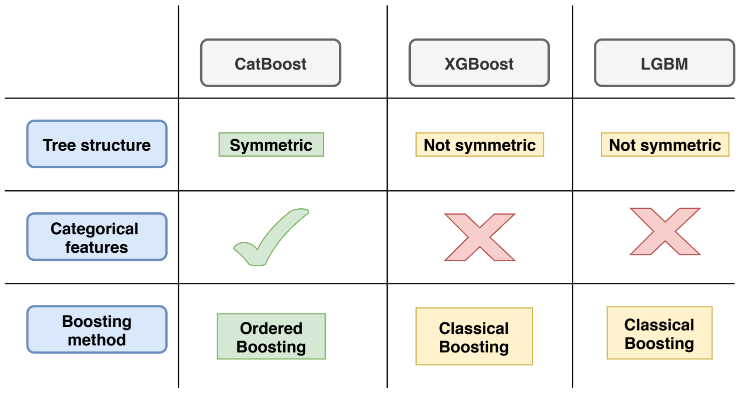


Figure 4.1: A comparison between gradient boost methods, i.e. CatBoost, XGBoost, and LGBM w.r.t., the way each method used to construct trees, tackling categorical features, and boosting method.

Tree structure. LGBM builds trees in a greedy fashion node by node and using this way; we can get a very deep, not symmetric tree (30). XGBoost builds trees layer by layer and then prunes them, which means it does not go deeper, and the resulting trees are not symmetric that lead to over-fitting. CatBoost uses full binary trees that are symmetric. This type of tree gives more simple predictors which means the algorithm is less go into over-fitting, and it also helps the algorithm to be more reliable to parameter changes.

Categorical handling. For categorical features, current algorithms either do not work with them at all such as XGBoost or deal with them in a not optimal way like LGBM, and the most common method to tackle categorical features is to use one-hot encoding. CatBoost handles categorical features in a very efficient way. It does a set of things w.r.t categorical features: (i) The first and most important thing in terms of quality is *statistics based on category and category plus label value.* In this step, it calculates a new set of numerical features based on the current existing categorical ones. Numerical features are calculated based on statistics related to category, e.g. the number of appearances of categories and at the same time based on label value, e.g. the average label value among objects that have the same category. For a binary classification it is the probability of having success in the given category for example (see figure 4.2) the new numerical feature value based on those statistics for object i is $P(1 \mid A) = \frac{3}{4}$, but this method leads to over-fitting. One way to overcome this problem, for each object, e.g. i averages are calculated differently in particular, for object number i objects that are before it, is used to calculate average statistics, and label of object i is never used during the calculation. For example, the object i has three objects before the given one at index 4, and the generated numerical feature is equal to $\frac{2}{3}$. If an object is the first one in the data set when we have zero by zero, then the *prior* value is needed, and for each feature, different priors are selected internally to make better qualities (7); (ii) Usage of several permutations to choose tree structure; (iii) Greedy constructed feature combinations, i.e. combine categorical features, e.g. *colour and animal* features become after combining them *blue_cat, or red_dog* and in many cases these combinations are meaningful; (iv) One hot encoding that works well if the data set has a small number of categories, e.g. gender (male or female).

Boosting method. In most gradient methods such as XGBoost and LGBM, they use *classical boosting* where leaf values (see equation 4.1) are calculated based on the average of all gradients in the leaf. This method is prone to over-fitting since all gradient estimates are always biased. CatBoost uses *ordered boosting* where gradient estimate is done for objects separately using a separate model, and accordingly, the estimate is unbiased (see equation 4.2). This method is *quadratic* per each iteration for memory and time, so instead of training n different models, we train $log(n)$ models (7).

$$leafValue = \sum_{i}^{n} \frac{g(approx(i), target(i)}{n} \tag{4.1}$$

$$leafValue = \sum_{i}^{doc} \frac{g(approx(i), target(i)}{docinthepast} \tag{4.2}$$

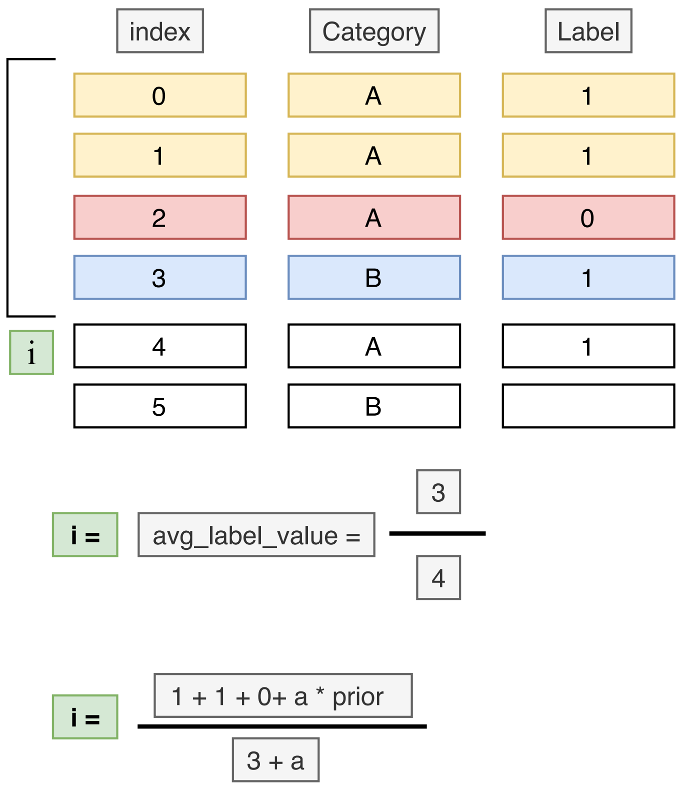


Figure 4.2: CatBoost tackles categorical features by calculating some statistics on the basis of the current and previous objects.

Following the general workflow of outcome-oriented predictive process monitoring in Chapter 3 section 3.1.1, we changed *sequence encoding* methods to keep all categorical features "as is" to let CatBoost algorithm tackle them during the learning process. Figure 4.3 shows the changes to all encoding methods w.r.t feature extraction.

4.2 Complex Sequence Encoding

Each case in the event log carries information about events that describe its *control flow* and each event is accompanied by information that describes the *data flow*. Most of the existing sequence encoding methods consider only the control flow variables (i.e. simple sequence encoding) from each trace, and therefore it loses *(lossy encoding)* information

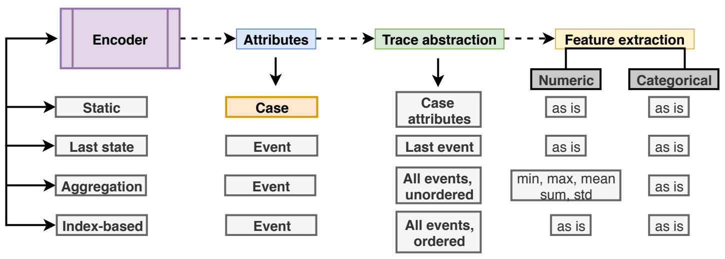


Figure 4.3: CatBoost with sequence encoding.

from historical event logs. In this thesis, we introduce a new complex sequence encoding approach. This method is a *lossless encoding* that captures information about control and data flow attributes with the help of *discrete wavelet transformation and auto-encoder neural networks* (see figure 4.4).

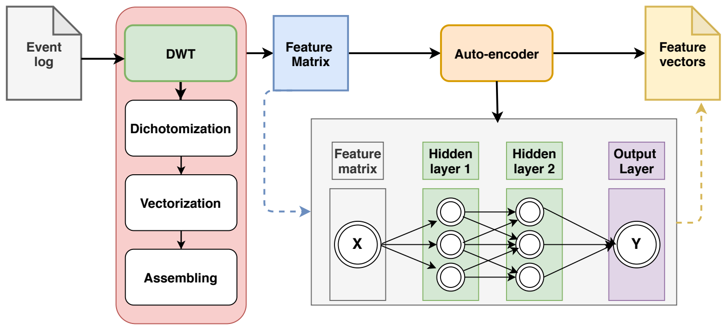


Figure 4.4: Our approach of building a new complex sequence encoding method based on discrete wavelet transformation (DWT) and autoencoder neural network.

4.2.1 Lossless encoding using Discrete Wavelet Transformation.

In this section, we present a complex sequence encoding technique on the basis of *discrete wavelet transformation* (31), which is considered as a *lossless* encoding method (32). In this approach, we encode each trace σ in the event log E into a group of vectors V with an identical length for each event in the trace. Each vector includes the corresponding *wavelet coefficients* for a distinct event. The significant perception is that for a given sequence of data, the *wavelet coefficients* expose accessible varieties within this sequence at distinctive analyses (32). Principally, each event log is composed of a collection of traces, and every trace contains many events, and each event has a specific timestamp associated with it, then wavelet coefficient is used to encode every timestamp in the event log. Formally, $\forall \sigma \in R, \sigma = \{e_1, e_2 \ldots, e_k\}$ then $DWT(\sigma) =< v_1, v_2, \ldots, v_k >$, where $|v_1| = |v_2| = \cdots = |v_k|$. This method was first introduced in process mining to find the variance between two event logs (32).

For any *vector space*, there is a tremendous amount of *basis matrices* as a specific matrix can be acquired using the help of others by a linear transformation (33), for example, *Haar transform matrix* which plays a significant role in analyzing time series (or sequential) data (34) such as event logs.

Definition 4.2.1. ***Haar transform matrix*** A Haar transform matrix is explained using the following recurrence equation as illustrated in (35) for a dimension of 2^n.

$$\mathbf{T}(n) = (\mathbf{T}(n-1) \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{U}(n-1) \otimes \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \mathbf{T}(0) = 1 \tag{4.3}$$

Where $\mathbf{T}(n)$ is the *Haar transform* that contains a set of vectors of length 2^n, $U(n)$ is a unit matrix and $\otimes$ is the outer product operator. For example:

$$\mathbf{T}(1) = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \text{ and } \mathbf{T}(2) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & -1 \end{pmatrix}$$

Definition 4.2.2. ***Sequential data*** A Sequential data $(s_1, \ldots, s_j)$ is a sequence of data points ordered in time that is assumed to be real values taken at discrete intervals in time, e.g. from time $t = 1$ to $time = j$ as defined in (36).

Definition 4.2.3. ***Discrete wavelet transformation*** Discrete wavelet transformation (DWT) is the representation of time series with *Haar matrix* and *wavelet coefficient*.

$$\mathbf{DWT}(s) = \mathbf{T} \cdot \mathbf{c} \tag{4.4}$$

Where $\mathbf{T}$ is the Haar transform, and $\mathbf{c}$ is a column vector that contains the *wavelet coefficient*.

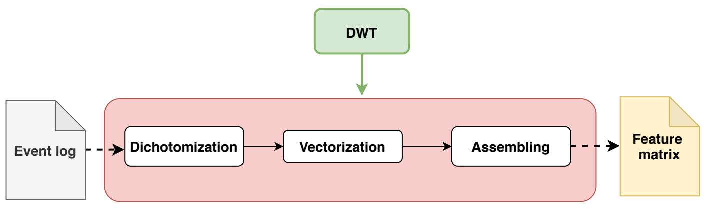


Figure 4.5: Discrete wavelet transformation (WDT) takes an event log as input and encodes it to a feature vector by doing three main steps, i.e. dichotomization, vectorization, and assembling.

4.2.2 DWT and autoencoder Neural network

Complex sequence encoding using *DWT* maps a trace into a multidimensional vector space. For each trace σ in an event log E, it generates sets of multidimensional features. This approach has many advantages over other sequence encoding methods (see Chapter 3) such as: (i) It is a lossless encoding method. (ii) Generated features are often uncorrelated (37, 38). Subsequently working with a wavelet coefficient is the same as working with raw data without any loss of information.

To encode event log using this approach into a feature vector we do three steps (see figure 4.5) for each trace: (i) *Dichotomization* which means extracting time series from a trace in the form of $\{0, 1\}$. *Zero* when event e does not exist in the input trace and *one* if it exists; (ii) *Vectorization* that encodes each time series into a vector through *DWT* which is discrete in time and scale, meaning it catches together frequency and location in time information (31); (iii) *Assembling* combine all generated vectors that capture wavelet coefficient information into one single vector for each trace.

Definition 4.2.4. ***dichotomization*** Let β be all possible events or actions in an event log E , and trace σ. we define function $B : \sigma \rightarrow |\beta|$ to map each input trace in E to a collection of $|\beta|$ sequential data of length $|\sigma|$. For example, $\sigma =< e_1, e_1, e_2, e_2 >$ with $\beta = \{e_1, e_2, e_3\}$, then $B(e_1, \sigma) = 1100$, and $B(e_2, \sigma) = 0011$, and $B(e_3, \sigma) = 0000$.

Definition 4.2.5. ***Vectorization*** we define function $V : B \rightarrow c$, that maps all time series into the relative *wavelet coefficients* where $B_i \in \{0, 1\}$. Formally, $c = T^{-1} \cdot B$, where T^{-1} is the inverse of Haar transform. Such as:

$$\mathbf{c}^{(i)} = \mathbf{T}^{-1} \cdot s^{(i)}$$

$$\mathbf{c}^{(1)} = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & -0.25 & -0.25 \\ 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0.5 & -0.5 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{c}^{(1)} = \begin{pmatrix} 0.5 \\ 0.5 \\ 0 \\ 0 \end{pmatrix}, \mathbf{c}^{(1)} = \begin{pmatrix} 0.5 \\ -0.5 \\ 0 \\ 0 \end{pmatrix}, \mathbf{c}^{(1)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Definition 4.2.6. ***Assembling*** We define W, i.e. a single vector for each trace σ by combining all vectors from definition 4.2.5 into one vector such as $W(\sigma) = (c^{(1)}, \ldots, c^{(n)})$. For example:

$$W(\sigma) =< 0.5, 0.5, 0, 0, 0.5, -0.5, 0, 0, 0, 0, 0, 0 >$$

Definition 4.2.7. ***Feature vector matrix*** Following the above definitions, we define a feature matrix M for each event log E, which rows belong to the assembled vectors of the relative traces from definition 4.2.6.

$$M(E) = \begin{pmatrix} W(\sigma_1) \\ \vdots \\ W(\sigma_n) \end{pmatrix}$$

Following the above definitions, we built a new complex sequence encoding method based on discrete wavelet transformation and time-series encoding by mapping each event log into a feature vector matrix whose rows are related to wavelet coefficients for each trace in the event log. This encoding method has two drawbacks compared to other encoding methods (see Chapter 3): (i) The time complexity in the worst case is $O(n^3)$ w.r.t the largest business case in the event log. Accordingly, for large datasets, this encoding method takes much time just to encode historical events into a numerical feature vector; (ii) secondly, the generated feature matrix is a *highly sparse multidimensional matrix*. To overcome the latter drawback, we can select a subset from the feature matrix based on the importance or apply one of the dimension reduction techniques. We decided to use an *auto-encoder* neural network to reduce the dimensionality of the feature matrix and to retain all information of the historical event logs.

Auto-encoder is an unsupervised method that tries to remove noise or unnecessary information from the original data. The idea of a neural network encoder is to compress the information that exists in the feature matrix with high dimensional space into a low dimensional space. After encoding each trace in the original event log using DWT, we pass it to the auto-encoder method (see figure 4.4) that compresses it into a lower dimension.

Following the general workflow of outcome-oriented predictive process monitoring in Chapter 3 section 3.1.1, we changed *sequence encoding* methods to add our proposed complex sequence encoding method using DWT*, and auto-encoders* Figure 4.6 shows the changes to all encoding methods w.r.t Sequence encoding.

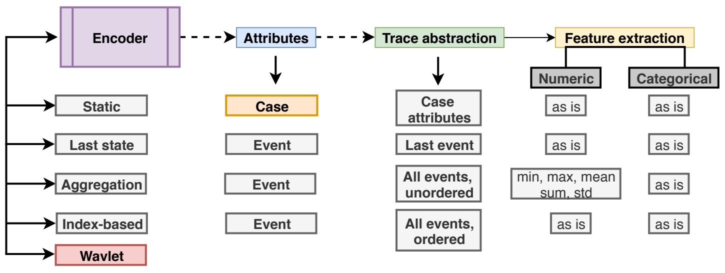


Figure 4.6: Taxonomy of existing and proposed (i.e. wavelet) sequence encoding methods for outcome-oriented PPM.

4.3 Inter-case features

In recent predictive process monitoring methods, the outcome mainly comes from the execution of the current running business process (i.e. *intra-case features*), on the other hand, in real-life systems the outcome does not depend only on the running case, but it depends on features that come from the concurrent execution of all business cases (i.e. *inter-case features*). For example, in many scenarios ongoing cases compete with each other for resources, then the number of ongoing cases that share the same resource may be a bottleneck to predict the outcome for an incomplete business case. In this section, we introduce new inter-case features that improve the outcome-oriented PPM in w.r.t the prediction accuracy.

In (6) they introduced a two-dimensional method for feature encoding, where the first dimension depends on the intra-case features, and the second one relies on the inter-case features. For the inter-case dimension, they proposed two different approaches: (i) *A knowledge-driven encoding (KDE)*, that relies on the previous knowledge about cases; (ii) *A data-driven encoding (DDE)*, that defines cases automatically from event logs employing proximity metrics.

In this thesis, we propose three different categories of inter-case features (see figure 4.7): (i) *Workload*; (ii) *Demand intensity*; (iii) *Temporal context* with an experimental evaluation executed on 20 event logs.

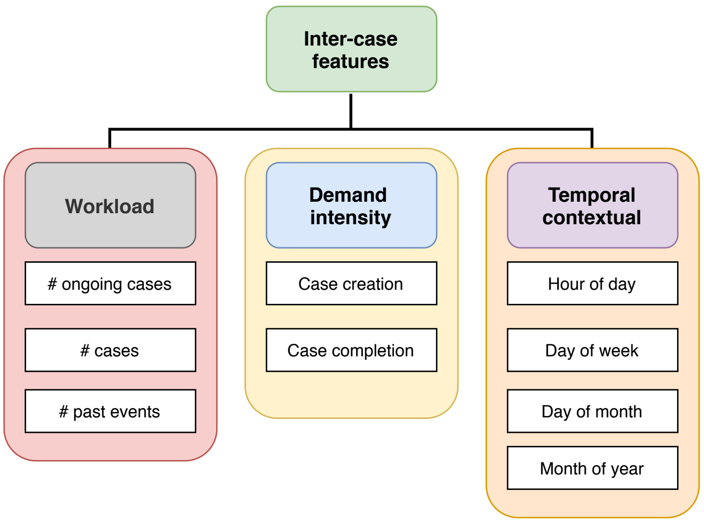


Figure 4.7: Categorization of inter-case features that we propose in this thesis, i.e. workload, demand intensity, and temporal contextual features.

Workload features. To improve the overall performance of business processes, we need to maximize the utilization of resources that we have in our system. Accordingly, monitoring and managing workload is crucial to any business process. In this context, we define three different inter-case features that capture information about the workload. (i) The number of business cases that have started and not finished; (ii) The number of activities that happened in the last x minutes or hour; (iii) Let C be an ongoing case, and e_c be the last event observed in C we define an inter-case feature that captures information about the number of other ongoing cases where the last event is e_c.

Demand intensity. The purpose of demand intensity analysis is demand prediction, and therefore, in terms of determining a proper resource to work on a specific action or activity, we need to monitor the intensity of business process creation and completion. to address this point, we define tow different features that capture information about the request intensity. (i) *Case creation intensity* that carries information about How many new cases were created since the current case started (divided by the number of seconds since the current case was created). (ii) *Case completion intensity*, on the other hand, capture information about how many cases have completed since this case was created divided by the number of seconds since the current case was created.

Temporal context Temporal information extraction has grown in the area of outcome-oriented PPM, and it is meant to capture circadian cycles, i.e. current time's hour of the day $(0 - 23)$, and day of the week when the last event in the current case occurred $(1 - 7)$. In this context, we defined four different inter-case features: (i) Hour of the day; (ii) Day of the week; (iii) Day of the month; (iv) Month of the year;

4.4 Summary

This Chapter addressed RQ1 by giving a complete description of our three different proposed approaches to improve the outcome-oriented predictive process monitoring tasks. In the next Chapter, we start by setting experiments to evaluate proposed methods in comparison to currently existing techniques.

5

Experimental Setting and Results

In Chapter 4 we introduced three different methods (i.e. CatBoost classifier, wavelet complex sequence encoding, and inter-case features) to improve the existing methods of outcome-oriented predictive process monitoring which guides us to establish the subsequent research questions.

RQ2 To what extent each of the three methods introduced in Chapter 4 enhances the performance of classifiers for PPM, relative to the baseline methods introduced in (5)?

RQ3 To what extent is the performance of the three methods introduced in Chapter 4 dependent on the choice of classification technique or sequence encoding method?

In this Chapter, we focus on answering the above research questions by making an analogous experimental evaluation of proposed and existing outcome-oriented PPM techniques (shown in Figure 5.1). To conduct our experiments, we used an open-source PPM environment in Python (5) and extended it to include our newly proposed methods. We used Python 3.6.3, Pandas 0.25.3, and Scikit-learn packages (39) to execute all experiments on the same resources that have been used in (5).

Within the rest of this chapter, we begin with presenting the assessment event logs, then specifying the assessment methods and end with showing the outcomes of this analysis.

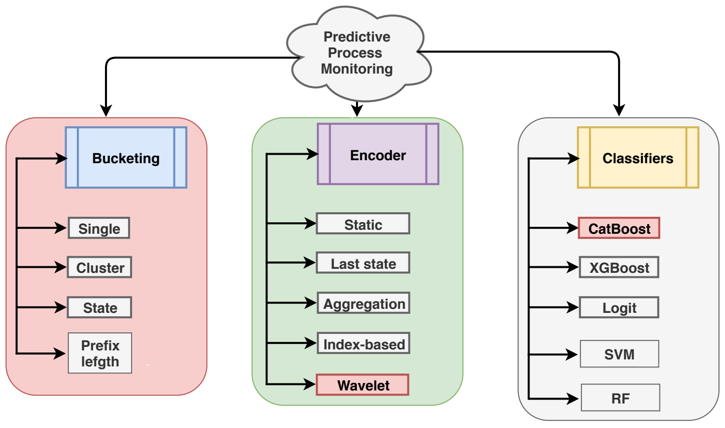


Figure 5.1: Existing and proposed sequence encoding, trace bucketing, and classification methods for outcome-oriented predictive process monitoring.

5.1 Datasets

Experiments are derived from 7 real-life historical event logs that include 4 diverse application domains: *banking* (event logs *BPIC2012* and *BPIC2017*), *government* (*BPIC2015* and *Traffic fines*), *manufacturing* (*Production*), *healthcare (BPIC2011, Sepsis.* All these datasets are publicly available and accessible from the 4TU Centre for Research Data (40). These logs contain dynamic event features, and static case features with 20 different outcome prediction tasks (5) where the final outcome of the case is defined in different approaches based on the business owner's objectives: "*bpic2012_A, bpic2012_C, bpic2012_D, bpic2017_A, bpic2017_C, bpic2017_R, bpic2015_1, bpic2015_2, bpic2015_3, bpic2015_4, bpic2015_5, traffic, production, bpic2011_1, bpic2011_2, bpic2011_3, bpic2011_4, sepsis_1, sepsis_2, and sepsis_3.*" In the coming paragraphs, we explain these logs, with the outcome-oriented predictive task in detail.

Table 5.1. shows some general statistics about the 20 event logs, namely the number of traces, min/mid/avg length of traces, number of case and event attributes, etc. This summarization can help when comparing different outcome-oriented predictive process monitoring methods to each other. The datasets have a wide range of characteristics and originate from many different domains.

Table 5.1: Event logs (Datasets) characteristics

dataset	# cases	min length	med length	max length	# distinct activities	# case variables	# event variables	# case cat values	# event cat values
bpic2011_1	1140	1	25.0	1814	193	6	14	961	290
bpic2011_2	1140	1	54.5	1814	251	6	14	994	370
bpic2011_3	1121	1	21.0	1368	190	6	14	886	283
bpic2011_4	1140	1	44.0	1432	231	6	14	993	338
bpic2015_1	694	2	42.5	101	380	17	12	17	432
bpic2015_2	753	1	55.0	132	396	17	12	7	429
bpic2015_3	1325	3	42.0	124	380	18	12	17	428
bpic2015_4	577	1	42.0	82	319	15	12	9	347
bpic2015_5	1051	5	50.0	134	376	18	12	8	419
production	220	1	9.0	78	26	3	15	37	79
sepsis_1	754	5	14.0	185	14	24	13	195	38
sepsis_2	782	4	13.0	60	15	24	13	200	40
sepsis_3	782	4	13.0	185	15	24	13	200	40
bpic2012_A	4685	15	35.0	175	36	1	10	0	98
bpic2012_C	4685	15	35.0	175	36	1	10	0	99
bpic2012_D	4685	15	35.0	175	36	1	10	0	98
bpic2017_A	31413	10	35.0	180	26	3	20	13	194
bpic2017_C	31413	10	35.0	180	26	3	20	13	194
bpic2017_R	31413	10	35.0	180	26	3	20	13	194
traffic	129597	2	4.0	20	10	4	14	53	172

1. *BPIC 2012* Is an event log started in the 2012 Business Process Intelligence Challenge (BPIC) and includes records related to a loan application process in a large financial institution. The process includes three different subprocesses: (i) process that records the loan application state; (ii) process that records the loan offer; (iii) process that records the application work items. In this thesis, we deal with a process that tracks the state of the offer. The outcome-oriented predictive task in this event log is if the offer is accepted, declined, or cancelled. This event log is split into three datasets, and each one contains a binary label (e.g. declined vs not declined) to consider it as a binary classification problem.

2. *BPIC 2017* This event log comes from the 2017 BPIC and includes records related to the loan application process in a Dutch financial institution which is the same as BPIC 2012. This version of loan application logs is larger and cleaner than BPIC2012. Similarly to the previous log, BPIC 2017 log is split into three datasets, and each one contains a binary label (e.g. accepted vs not accepted) to

consider it as a binary classification problem.

3. *BPIC 2015* This event log records and combines event logs from 5 Dutch cities w.r.t a building permit application process. It is split into five different logs, one for each city and the outcome that we aim to predict for predictive monitoring is for each trace if an LTL rule (41) is violated or fulfilled (22, 42).

4. *Traffic fines* This event log records processes related to traffic fines management, and it contains information about fine notification and at the same time, partial repayments. It comes from an Italian police unit. We define the final outcome on the basis of whether the penalty is returned complete or not.

5. *Production* Is an event log that reports events of the production process. Every case tracks data about the actions, process operators, and resources contributed to items production. We define the final outcome of an ongoing case on the basis of the number of discarded product orders is more significant than 0 or not.

6. *BPIC 2011* This event log shows information about the medical history of patents of a Dutch Academic Hospital, and it comes from the first BPIC. The predictive task w.r.t this log is the same as BPIC 2015 that is for each trace if an LTL rule (41) is violated or fulfilled.

7. *Sepsis* Is an event log that holds information about sepsis events from a Dutch hospital. Each trace outlines the pathway for a patient within this clinic. This data set is split into three logs based on the outcome for cases, e.g. whether the patient is admitted to an intensive care unit or not.

5.2 Experimental setup

In this section, we present our technique to divide the historical event logs to a train, test, plus evaluation sets onward with describing each evaluation measure that we applied to end with optimizing the hyper-parameters of the proposed methods.

5.2.1 Data Preprocessing

Event logs preprocessing is one of the important and essential steps before building any predictive model (43). To achieve the highest possible prediction accuracy; data preprocessing for all logs should be done beforehand. Firstly we filter out incomplete traces as well as traces that have not been ordered from their beginning. Secondly, timestamp features are parsed from the prior actions in all traces. Thirdly, For categorical features, the number of levels (i.e. distinct values) are calculated and consider only the most frequent values and other infrequent values are marked as other. Finally, event logs are recorded by enterprise systems, and it could be that some missing data exist. This happens because of the reality that some variables are not relevant to a particular action. In this context, we searched for the nearest preceding event in the same trace where the variables were available and used its value instead.

5.2.2 Data splits

The outcome-oriented predictive process monitoring problem follows the same workflow of any machine learning problem (see Chapter 2). In order to *tune and evaluate* predictive models, we need to split original event logs into three different parts (see figure 5.2), two of them to train and tune predictive models, and the third acts as unseen data for the model to evaluate its power. In other words, we need to teach an algorithm on how to learn enough to predict cases related to the future along with avoiding *under-fitting* (i.e. learning less than enough) and *over-fitting* (learn more than enough). To achieve that, we need to split data into training, and testing sets properly. There are two methods in machine learning to split original data (44): (i) *train-test split*, and (ii) *cross-validation*.

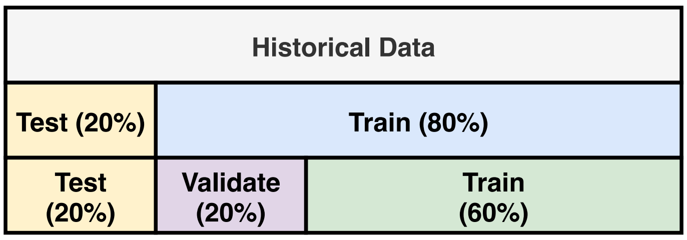


Figure 5.2: Data splitting

In real life, we train a predictive model on historical data, and apply it to ongoing traces; accordingly, we use *temporal split* (see section 2.2) to simulate real-life situations.

Particularly, business cases are arranged using their timestamps, and the opening 80% is employed for training (i.e. teaching models how to learn from data), and the rest is used to evaluate model performance. In other words, a machine learning classifier is trained and optimized by using all business cases that began before a presented time, and the evaluation is performed on instances that begin afterwards. This approach has a drawback which is, the overlapping between events in training cases; accordingly, training cases are cut in a way that events that overlay with evaluation instances are dropped to overcome this problem.

5.2.3 Evaluation Metrics

As discussed in Chapter 2 in section 2.3, a predictive model should be accurate and make a prediction as early as possible, along with predicting a reasonable amount of time. We evaluate the power of the model using three different metrics: accuracy, earliness, and efficiency.

1. *Accuracy* AUC has many advantages to use as a measure of the prediction accuracy, (i) it is unbiased towards a specific class in imbalanced datasets; (ii) Its a threshold-independent; Accordingly, we used AUC to report model accuracy. To make sure there is no biased viewpoint of results, we reported F-score with the default threshold of 0.5 as well.

2. *Earliness.* A general strategy to evaluate earliness of forecasts is for each incoming event we calculate the accuracy of prediction based on that event or at fixed time intervals. Meanwhile, the quicker we get the wanted value of accuracy, the more trustworthy the model w.r.t the earliness.

3. *Efficiency.* To estimate the execution time for predictive models, we distinguish between the training (or offline) and prediction (or online) phases. In the *training* phase, the total time is calculated based on the total amount of time that is needed for extracting prefixes from the original event log, prefix bucketing, and prefix encoding. When in the *prediction step*, we consider that a forecast is generated virtually immediately since, in real life, forecasts are demanded in real-time.

5.2.4 Baselines

We compare our proposed methods against work submitted by (5) as we consider it as a baseline for our work. We used a different combination of existing sequence encoding methods, bucketing types, and classifiers for outcome-oriented predictive process monitoring (see figure 5.1).

We used four different classification algorithms from existing predictive monitoring methods for the experiments and to compare them with the proposed predictive model (i.e. CatBoost): (i) gradient boosting method (XGBoost) (6, 45), random forest (RF) (16, 46), support vector machines (SVM), and logistic regression (Logit). For bucketing methods that are based on clustering, we used $k-means$ that is widely used in different domains.

Table 5.2: Hyper-parameter optimization using TPE

Method	Parameter	Distribution	Values
CatBoost	Learning rate	uniform	z ∈ [0.01, 0.8]
	One hot encoding max size	uniform integer	z ∈ [4, 255]
	Max tree depth	uniform integer	z ∈ [6, 16]
	Colsample bytree	uniform	z ∈ [0.5, 1]
	Bagging temperature	uniform	z ∈ [0, 100]
	Scale pos weight	uniform	z ∈ [1, 16]
	L2 leaf regularization	log uniform	z ∈ [0, 10]
	number of estimators	uniform integer	z ∈ [500, 1000]
XGBoost	Learning rate	Uniform	z ∈ [0, 1]
	Subsample	Uniform	z ∈ [0, 1]
	Max tree depth	Uniform integer	z ∈ [0, 1]
	Colsample bytree	Uniform	z ∈ [0, 1]
	Min child weight	Uniform integer	z ∈ [0, 1]
	number of estimators	uniform integer	z ∈ [500, 100]
RF	Max features	Uniform	z ∈ [0, 1]
Logit	Inverse of regularization strength (C)	Uniform integer	2^z , z ∈ [−15, 15]
SVM	Penalty parameter (C)	Uniform integer	2^z , z ∈ [−15, 15]
	Kernel coefficient (gamma)	Uniform integer	2^z , z ∈ [−15, 15]
K-means	Number of clusters	Uniform integer	z ∈ [2, 50]

5.2.5 Hyper-parameter Optimization

Within each predictive model (or classifier) and clustering algorithms (e.g. k-means) there are two types of parameters, first type (i.e. *fitted parameters*) that model can learn during the training process, e.g. *weights* of features, and the other type (i.e. *hyper-parameters*) where we need to adjust and tune during the training process to achieve the best possible performance, e.g. *max_depth, and learning rate.* To tune (or optimize) the hyper-parameter we moreover split the training data to 60% for training and 20% for tuning (or validation) using *k-fold cross validation* when $k = 3$. The training data is used to train models using combinations of hyper-parameters, where the validation data is applied to evaluate the performance of various hyper-parameter combinations. We used the TPE algorithm to select the best parameter for our model. Table 5.2 shows the bounds and the sampling numbers (i.e. search space) for all of the parameters provided as information to the optimizer.

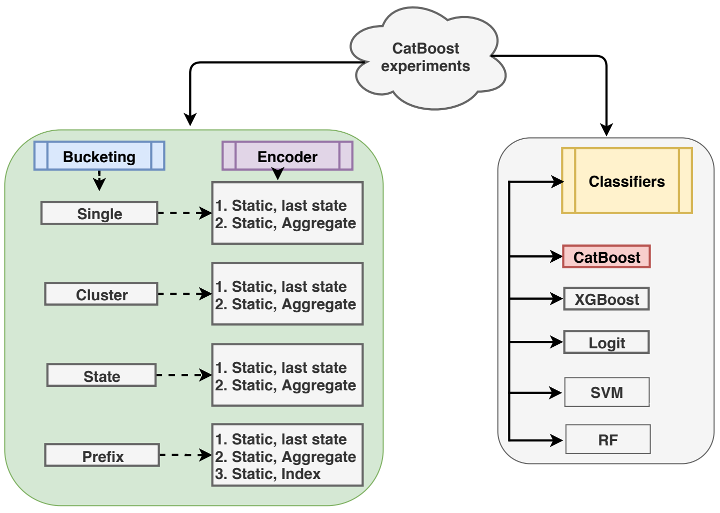


Figure 5.3: Catboost outcome-oriented PPM experiments.

5.3 Results: accuracy and earliness

In this section, we provide experiments results in terms of the first two evaluation metrics, i.e. *accuracy and earliness* for our proposed methods to improve the outcome-oriented predictive process monitoring problem.

5.3.1 Contribution 1: (CatBoost results)

in this part, we executed different experiments based on a different combination between *bucketing, encoding, and classification* methods as shown in figure 5.3

Overall prediction scores (AUC and F-score) are estimated by calculating the score independently for all prefixes with different length for a given case; after that, we calculate the weighted average of all collected prediction scores. Weights are corresponding to the number of prefixes employed to calculate the initial prediction scores, and it confirms that scores are controlled fairly by all prefixes, rather than influencing by largest prefixes. Table 5.3 show results of CatBoost classifier in terms of AUC and F-score, and for XGBoost, SVM, RF, and Logit results are shown in Tables B.1, B.4, B.3, and B.2.

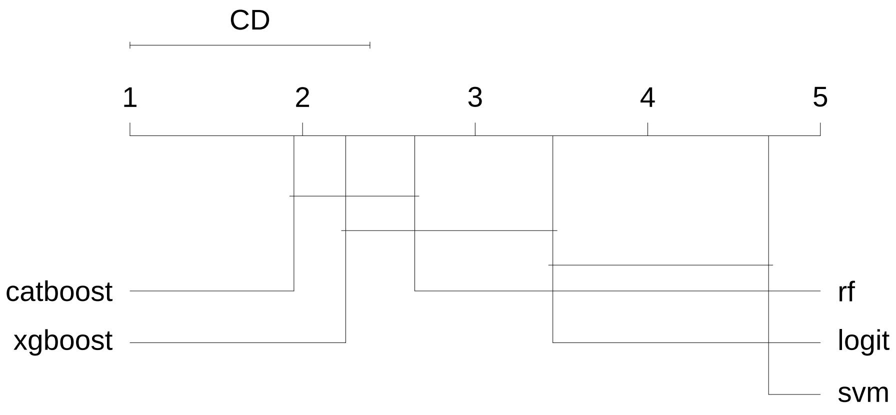


Figure 5.4: Critical difference diagram based on the highest AUC values using Nemenyi test shows there is no statistical difference (at $pvalue < \alpha$) between connected methods using a straight line such as CatBoost, XGBoost, and Random Forest.

Results show that CatBoost has some improvements against all other classification methods since it gives the best (or shared best) prediction score (i.e. AUC) in 10 out of 20 event logs, followed by Xgboost, which shows the best AUC (or shared best) in 5 event logs, Logit scores the best (or shared best) AUC in 1 event logs. RF scores the

Table 5.3: Overall AUC (F-score) for **CatBoost**

	catboost					
	bpic2012_a	bpic2011_4	bpic2012_d	bpic2012_c	production	bpic2015_4
single_agg	0.62 (0.65)	**0.92** (**0.82**)	0.59 (0.19)	0.64 (0.42)	0.68 (0.58)	0.72 (0.14)
state_laststate	0.67 (0.68)	0.86 (0.73)	**0.62** (**0.21**)	0.71 (0.5)	0.64 (0.53)	0.74 (0.32)
cluster_laststate	0.66 (0.64)	0.9 (0.78)	0.6 (0.22)	0.68 (0.46)	0.65 (0.56)	0.7 (0.21)
prefix_laststate	0.64 (0.66)	0.91 (0.79)	0.6 (0.24)	0.69 (0.43)	0.67 (0.58)	0.66 (0.3)
prefix_agg	0.64 (0.66)	0.9 (0.79)	0.58 (0.17)	0.67 (0.39)	0.65 (0.56)	0.68 (0.28)
state_agg	0.68 (0.7)	0.84 (0.72)	**0.62** (**0.21**)	0.7 (0.47)	0.68 (0.6)	0.71 (0.19)
cluster_agg	0.68 (0.69)	**0.92** (**0.79**)	0.61 (0.18)	0.7 (0.5)	0.68 (0.59)	0.65 (0.23)
prefix_index	0.59 (0.6)	**0.92** (**0.8**)	0.59 (0.6)	0.55 (0.35)	0.71 (0.67)	0.71 (0.26)
single_laststate	0.64 (0.69)	**0.92** (**0.82**)	0.59 (0.19)	0.64 (0.42)	0.61 (0.55)	0.72 (0.14)

	catboost					
	bpic2015_2	traffic	sepsis_3	sepsis_2	bpic2011_1	bpic2015_3
single_agg	0.92 (0.43)	0.7 (0.74)	0.74 (0.32)	0.53 (0.03)	0.77 (0.69)	0.66 (0.3)
state_laststate	0.89 (0.4)	0.69 (0.72)	0.73 (0.37)	0.36 (0.03)	0.86 (0.76)	**0.67** (**0.32**)
cluster_laststate	0.76 (0.56)	0.68 (0.73)	0.73 (0.32)	0.53 (0.03)	0.86 (0.72)	**0.67** (**0.3**)
prefix_laststate	0.95 (0.43)	0.71 (0.74)	**0.76** (**0.36**)	0.53 (0.03)	0.88 (0.77)	0.63 (0.24)
prefix_agg	0.94 (0.58)	0.68 (0.69)	0.72 (0.27)	0.53 (0.03)	0.91 (0.76)	0.61 (0.25)
state_agg	0.89 (0.42)	0.67 (0.67)	0.73 (0.03)	0.36 (0.03)	0.88 (0.75)	0.64 (0.19)
cluster_agg	0.53 (0.98)	0.71 (0.73)	0.63 (0.14)	0.53 (0.03)	0.88 (0.71)	0.64 (0.31)
prefix_index	0.8 (0.46)	**0.74** (**0.21**)	0.73 (0.27)	0.88 (0.62)	0.83 (0.77)	0.6 (0.3)
single_laststate	0.86 (0.58)	0.69 (0.74)	0.71 (0.27)	0.53 (0.03)	0.87 (0.7)	0.66 (0.3)

	catboost					
	bpic2015_1	bpic2011_2	bpic2011_3	bpic2017_a	bpic2015_5	bpic2017_r
single_agg	0.76 (0.36)	0.91 (0.84)	0.95 (0.83)	0.71 (0.75)	0.73 (0.53)	0.8 (0.45)
state_laststate	0.77 (0.53)	0.64 (0.79)	0.91 (0.8)	0.82 (0.74)	0.66 (0.43)	0.81 (0.49)
cluster_laststate	0.74 (0.48)	0.89 (0.86)	0.94 (0.81)	0.83 (0.72)	**0.74** (**0.56**)	0.8 (0.48)
prefix_laststate	0.74 (0.39)	0.94 (0.87)	**0.97** (**0.87**)	0.81 (0.79)	0.7 (0.49)	0.8 (0.49)
prefix_agg	0.71 (0.4)	0.94 (0.86)	0.95 (0.83)	0.72 (0.76)	0.73 (0.51)	0.81 (0.5)
state_agg	0.78 (0.48)	0.58 (0.77)	0.91 (0.8)	0.72 (0.74)	0.67 (0.52)	**0.82** (**0.52**)
cluster_agg	0.68 (0.6)	0.52 (0.98)	0.94 (0.79)	0.71 (0.8)	0.73 (0.49)	0.8 (0.48)
prefix_index	0.74 (0.34)	0.9 (0.89)	0.96 (0.86)	0.63 (0.6)	0.72 (0.16)	0.58 (0.18)
single_laststate	0.82 (0.52)	0.93 (0.86)	0.95 (0.85)	0.84 (0.75)	0.73 (0.53)	0.8 (0.46)

	catboost	
	bpic2017_c	sepsis_1
single_agg	0.82 (0.8)	0.53 (0.03)
state_laststate	0.82 (0.76)	0.46 (0.11)
cluster_laststate	0.81 (0.78)	**0.57** (**0.2**)
prefix_laststate	0.81 (0.75)	0.48 (0.07)
prefix_agg	0.82 (0.79)	0.45 (0.07)
state_agg	0.82 (0.79)	0.5 (0.15)
cluster_agg	0.81 (0.78)	0.51 (0.1)
prefix_index	0.56 (0.62)	0.55 (0.13)
single_laststate	0.83 (0.8)	0.56 (0.09)

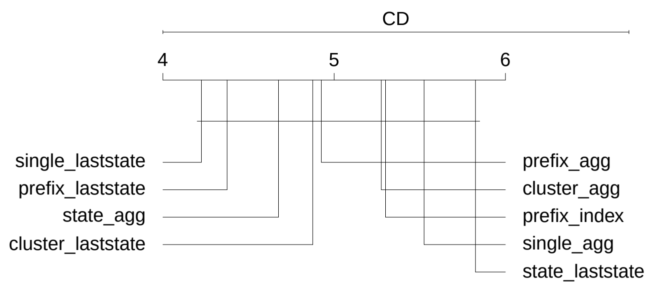
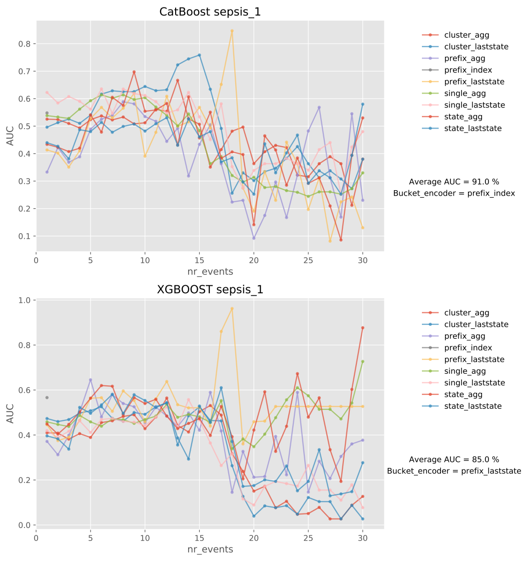


Figure 5.5: Critical difference diagram between all trace bucketing and sequence encoding methods shows there is no statistical difference (at $pvalue < \alpha$) between all of them when we use CatBoost classifier.

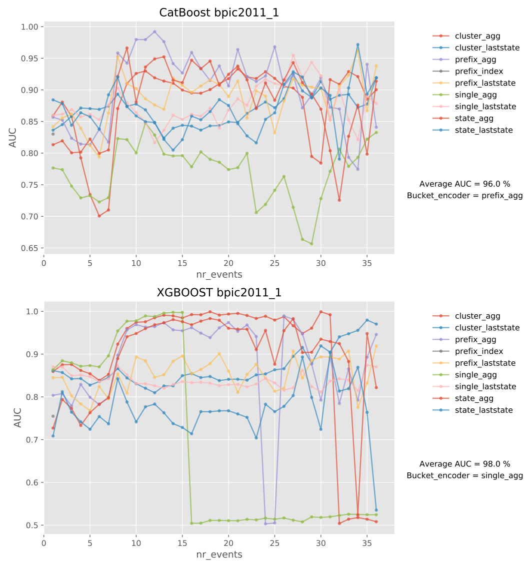
best (or shared best) AUC in 5 logs. SVM, on average, does not give the equivalent level of scores as the other methods.

We used *A/B testing* to quantify the difference between all classifiers in terms of the performance. We employed the *Nemenyi* test as presented by (47)) to plot the *critical difference diagram* as a statistical method to test whether the difference between all methods is significant or not with $\alpha = 0.05$. Critical difference diagram in figure 5.4 show that the best performer classifier is CatBoost with an average level of 1.95; however, there is no statistical difference between it, XGBoost, and RF.

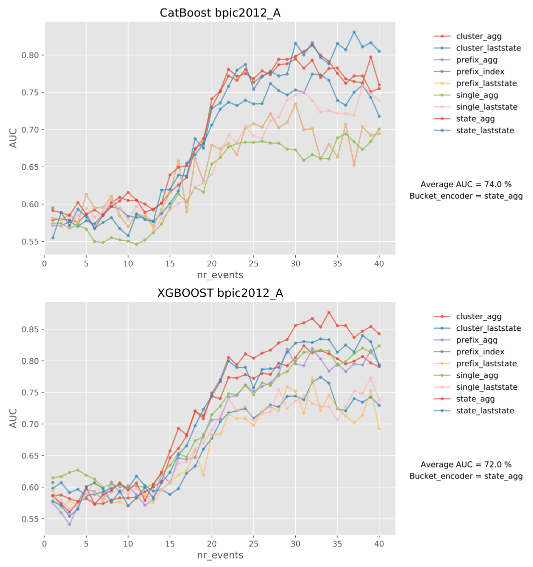
For sequence encoding methods, and trace bucketing, we can see in Table 5.4a different combinations between them and the number of event logs that each combination worked the best. From these results, we can not say that there is a specific bucketing, and encoding methods worked the best for CatBoost classifier. These results agree with the critical difference diagram, that is shown in figure 5.5. Figures 5.6, 5.7, and 5.8, show a comparison between the best two classifiers, i.e. CatBoost and XGBoost based on the predicted accuracy in terms of AUC. All sub-figures contain information about the best AUC and the corresponding bucketing and encoding method while figures B.1, B.4, and B.7 in the Appendix show all other comparisons. This analysis gathers the answer to RQ2 (To what extent each of the three methods introduced in Chapter 4 enhances the performance of classifiers for PPM, relative to the baseline methods introduced in (5)?), and RQ3 (To what extent is the performance of the three methods introduced in Chapter 4 dependent on the choice of classification technique or sequence encoding method?).



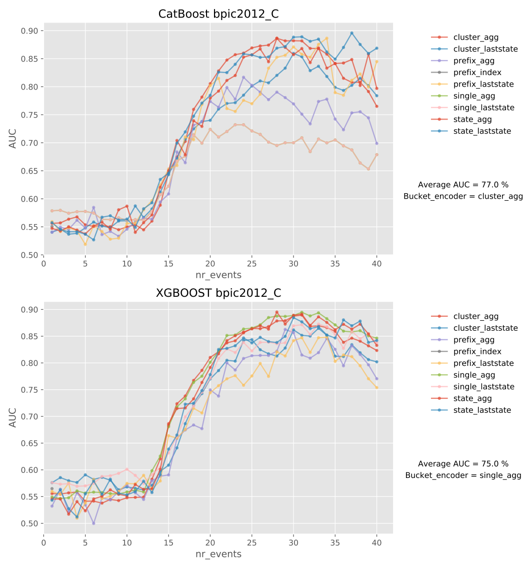
(a) CatBoost vs XGBoost sepsis



(b) CatBoost vs XGBoost bpic2011

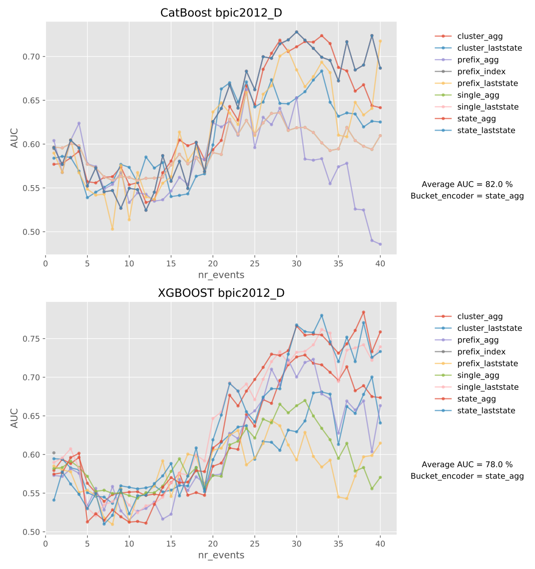


(c) CatBoost vs XGBoost bpic2012_A

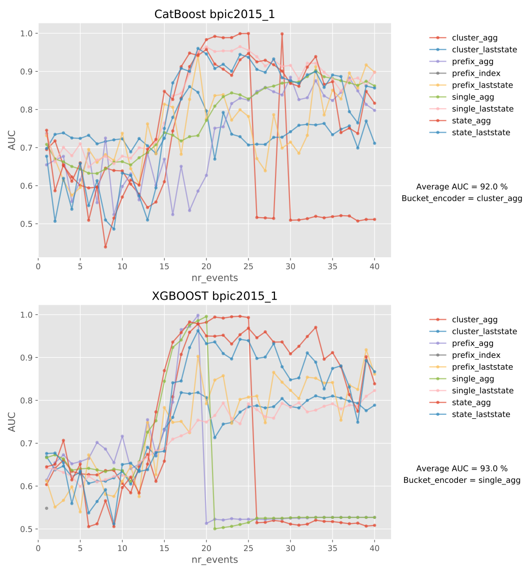


(d) CatBoost vs XGBoost bpic2012_C

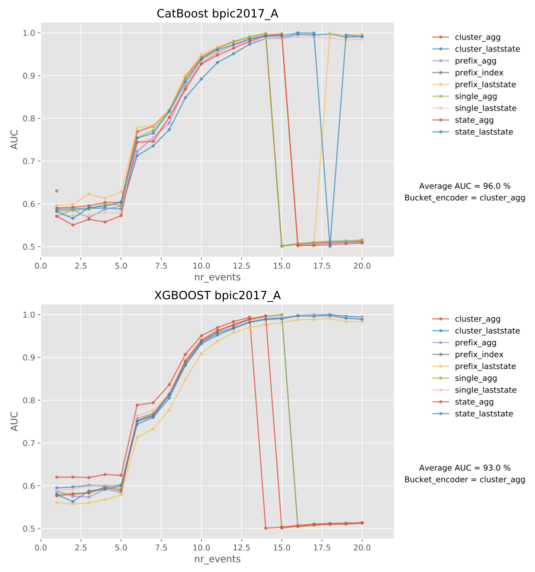
Figure 5.6: Comparison between CatBoost and XGBoost on all event logs



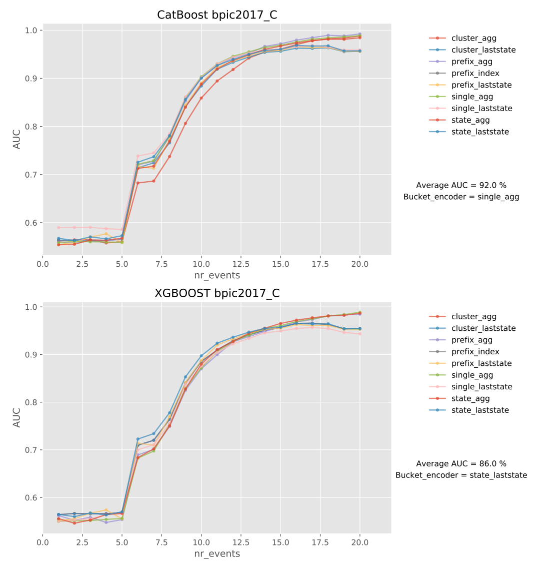
(a) CatBoost vs XGBoost bpic2012_D



(b) CatBoost vs XGBoost bpic2015

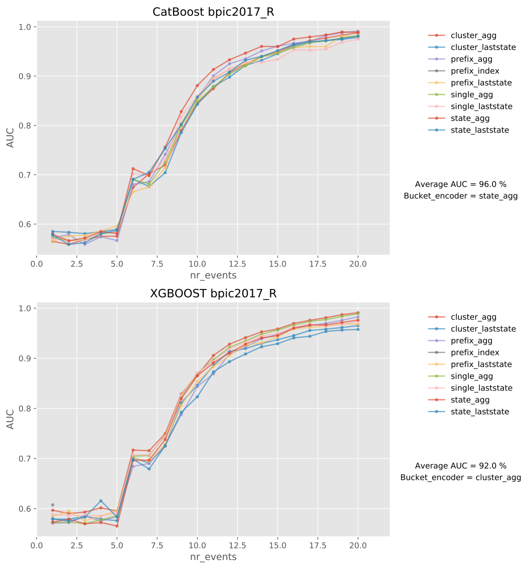


(c) CatBoost vs XGBoost bpic2017_A

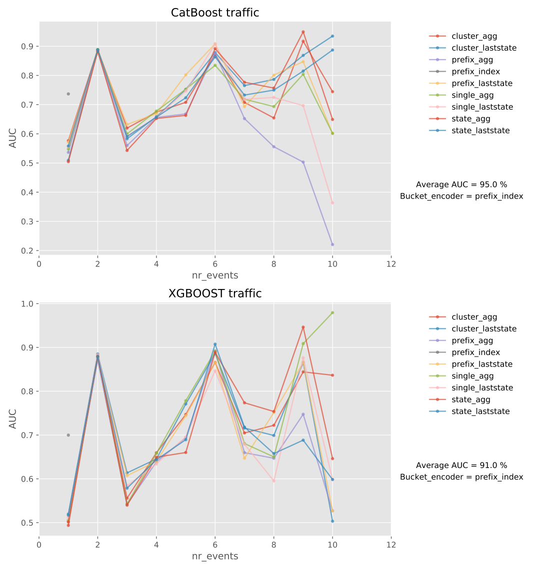


(d) CatBoost vs XGBoost bpic2017_C

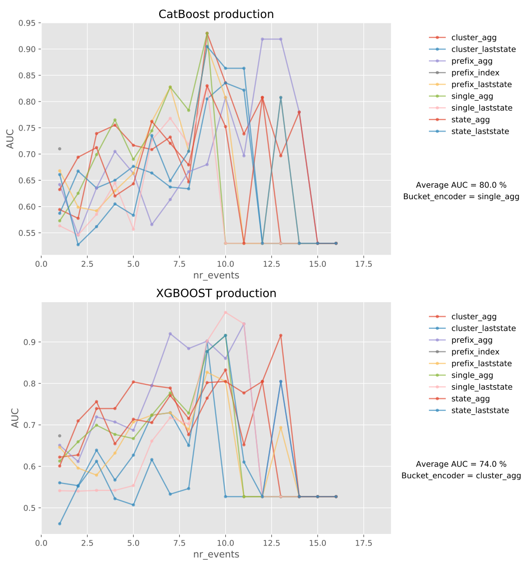
Figure 5.7: Comparison between CatBoost and XGBoost on all event logs (*continued*)



(a) CatBoost vs XGBoost bpic2017_R



(b) CatBoost vs XGBoost traffic



(c) CatBoost vs XGBoost Production

Figure 5.8: Comparison between CatBoost and XGBoost on all event logs (*continued*)

5.3.2 Contribution 2: Complex sequence encoding (Wavelet) results:

In this part, we added the complex sequence encoding using discrete wavelet transformation and auto-encoder method that we proposed to the experiments. Now we combine previous encoding methods with the newly added one, as shown in figure 5.9.

We followed the same strategy as we did in section 5.3.1 where we calculated all prediction scores across all prefix lengths, and after that, we calculated the average score for all predictions based on the number of tested prefixes for each case.

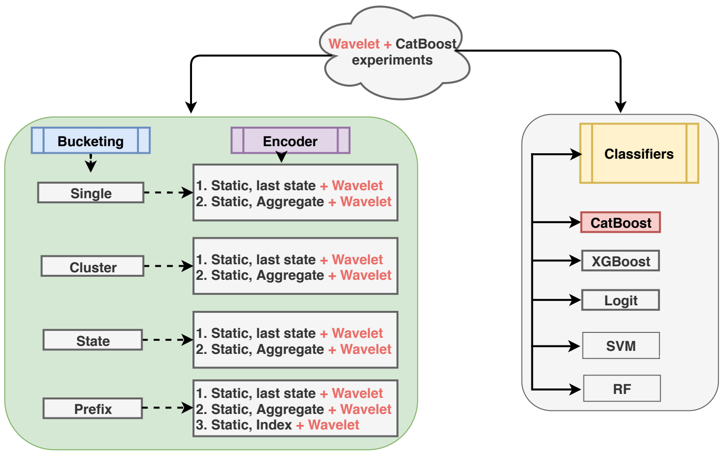

Wavelet + CatBoost experiments
Bucketing
Encoder
Single
Cluster
State
Prefix
1. Static, last state + Wavelet
2. Static, Aggregate + Wavelet
3. Static, Index + Wavelet
Classifiers
CatBoost
XGBoost
Logit
SVM
RF

Figure 5.9: Wavelet plus CatBoost outcome-oriented PPM experiments

Table 5.5 show collected results of CatBoost classifier with the wavelet encoding method in terms of AUC and F-score, and for XGBoost, SVM, RF, and Logit results are shown in Tables B.9, B.12, B.11, and B.10.

Results show that CatBoost has no improvements against all other classification methods since it gives the best (or shared best) prediction score (i.e. AUC) in 11 out of 20 event logs, followed by Xgboost, which shows the best AUC (or shared best) in 5 event logs, Logit scores the best (or shared best) AUC in 4 event logs. RF scores the best (or shared best) AUC in 2 logs.

Critical difference diagram in figure 5.10 show that the best performer classifier is CatBoost with an average level of 1.8; however, there is no statistical difference between it, XGBoost, Logit, and RF.

Table 5.4: Bucketing and sequence encoding methods

(a) CatBoost

Bucket_Encoding_method	# Data sets
cluster_agg_catboost	2
cluster_laststate_catboost	3
prefix_agg_catboost	2
prefix_index_catboost	4
prefix_laststate_catboost	4
single_agg_catboost	1
single_laststate_catboost	4
state_agg_catboost	3
state_laststate_catboost	3

(b) Wavelet plus CatBoost

Bucket_Encoding_Method	# Data sets
cluster_waveletAgg_catboost	7
cluster_waveletLast_catboost	2
prefix_waveletAgg_catboost	2
prefix_WaveletIndex_catboost	1
prefix_waveletLast_catboost	2
single_waveletAgg_catboost	5
single_waveletLast_catboost	2
state_waveletAgg_catboost	5
state_waveletLast_catboost	3

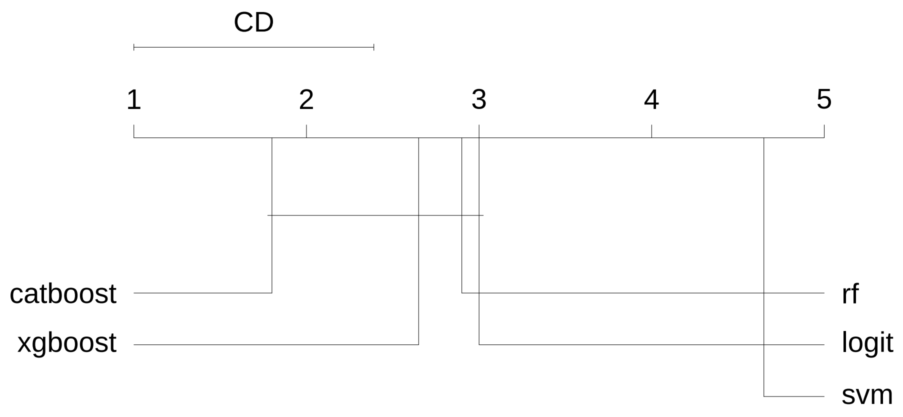


Figure 5.10: Critical difference diagram after adding wavelet encoding method based on the highest AUC values using Nemenyi test shows there is no statistical difference (at $pvalue < \alpha$) between connected methods using a straight line such as CatBoost, XGBoost, Random Forest, and Logistic Regression.

For sequence encoding and trace bucketing methods, we can see in Table 5.4b different combinations between them and the number of event logs that each combination worked the best with CatBoost classifier. From these results, we conclude that the *wavelet aggregate* method is the best across all combinations since it achieves the best score over 7 event logs.

Table 5.5: Overall AUC (F-score) for **Wavelet plus CatBoost**

	catboost					
	bpic2015_2	bpic2015_4	bpic2012_c	bpic2011_4	bpic2012_d	traffic
cluster_waveletAgg	0.56 (0.51)	0.68 (0.26)	0.73 (0.53)	**0.95** (**0.82**)	0.64 (0.21)	**0.74** (**0.76**)
prefix_WaveletIndex	0.79 (0.45)	0.7 (0.25)	0.54 (0.34)	0.91 (0.79)	0.58 (0.59)	0.73 (0.2)
prefix_waveletLast	0.95 (0.43)	0.66 (0.3)	0.69 (0.43)	0.91 (0.79)	0.6 (0.24)	0.71 (0.74)
state_waveletAgg	0.92 (0.45)	0.74 (0.22)	0.73 (0.5)	0.87 (0.75)	**0.65** (**0.24**)	0.7 (0.7)
single_waveletLast	0.87 (0.59)	0.73 (0.15)	0.65 (0.43)	0.93 (0.83)	0.6 (0.2)	0.7 (0.75)
single_waveletAgg	0.95 (0.46)	0.75 (0.17)	0.67 (0.45)	**0.95** (**0.85**)	0.62 (0.22)	0.73 (0.77)
state_waveletLast	0.9 (0.42)	0.75 (0.33)	0.72 (0.51)	0.87 (0.74)	0.63 (0.22)	0.7 (0.73)
prefix_waveletAgg	0.95 (0.59)	0.69 (0.29)	0.68 (0.4)	0.91 (0.8)	0.59 (0.18)	0.69 (0.7)
cluster_waveletLast	0.77 (0.57)	0.71 (0.22)	0.69 (0.47)	0.91 (0.79)	0.61 (0.23)	0.69 (0.74)

	catboost					
	bpic2012_a	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1
cluster_waveletAgg	0.71 (0.72)	0.67 (0.34)	0.97 (0.82)	0.55 (0.51)	0.56 (0.06)	0.62 (0.62)
prefix_WaveletIndex	0.58 (0.59)	0.59 (0.29)	0.95 (0.85)	0.89 (0.88)	0.87 (0.61)	0.73 (0.33)
prefix_waveletLast	0.64 (0.66)	0.63 (0.24)	0.97 (0.87)	0.94 (0.87)	0.53 (0.03)	0.74 (0.39)
state_waveletAgg	0.71 (0.73)	0.67 (0.22)	0.94 (0.83)	0.61 (0.8)	0.39 (0.06)	0.81 (0.51)
single_waveletLast	0.65 (0.7)	0.67 (0.31)	0.96 (0.86)	0.94 (0.86)	0.54 (0.04)	**0.83** (**0.53**)
single_waveletAgg	0.65 (0.68)	**0.69** (**0.33**)	**0.98** (**0.86**)	0.94 (0.87)	0.56 (0.06)	0.79 (0.39)
state_waveletLast	0.68 (0.69)	0.68 (0.33)	0.92 (0.81)	0.65 (0.8)	0.37 (0.04)	0.78 (0.54)
prefix_waveletAgg	0.65 (0.67)	0.62 (0.26)	0.96 (0.84)	**0.95** (**0.87**)	0.54 (0.04)	0.72 (0.41)
cluster_waveletLast	0.67 (0.65)	0.68 (0.31)	0.95 (0.82)	0.9 (0.87)	0.54 (0.04)	0.75 (0.49)

	catboost					
	bpic2015_5	sepsis_3	bpic2017_c	production	bpic2017_r	sepsis_1
cluster_waveletAgg	**0.76** (**0.52**)	0.66 (0.17)	0.71 (0.81)	0.71 (0.62)	0.73 (0.51)	0.54 (0.13)
prefix_WaveletIndex	0.71 (0.15)	0.72 (0.26)	0.55 (0.61)	0.7 (0.66)	0.57 (0.17)	0.54 (0.12)
prefix_waveletLast	0.7 (0.49)	0.76 (0.36)	0.81 (0.75)	0.67 (0.58)	0.8 (0.49)	0.48 (0.07)
state_waveletAgg	0.7 (0.55)	0.76 (0.06)	0.72 (0.82)	0.71 (0.63)	0.73 (0.55)	0.53 (0.18)
single_waveletLast	0.74 (0.54)	0.72 (0.28)	**0.84** (**0.81**)	0.62 (0.56)	0.81 (0.47)	0.57 (0.1)
single_waveletAgg	**0.76** (**0.56**)	**0.77** (**0.35**)	0.73 (0.83)	0.71 (0.61)	0.76 (0.48)	0.56 (0.06)
state_waveletLast	0.67 (0.44)	0.74 (0.38)	0.83 (0.77)	0.65 (0.54)	0.82 (0.5)	0.47 (0.12)
prefix_waveletAgg	0.74 (0.52)	0.73 (0.28)	0.81 (0.8)	0.66 (0.57)	0.79 (0.51)	0.46 (0.08)
cluster_waveletLast	0.75 (0.57)	0.74 (0.33)	0.82 (0.79)	0.66 (0.57)	0.81 (0.49)	**0.58** (**0.21**)

	catboost	
	bpic2011_1	bpic2017_a
cluster_waveletAgg	0.91 (0.74)	0.66 (0.73)
prefix_WaveletIndex	0.82 (0.76)	0.62 (0.59)
prefix_waveletLast	0.88 (0.77)	0.81 (0.79)
state_waveletAgg	0.91 (0.78)	0.67 (0.73)
single_waveletLast	0.88 (0.71)	0.8 (0.76)
single_waveletAgg	0.8 (0.72)	0.66 (0.66)
state_waveletLast	0.87 (0.77)	0.69 (0.75)
prefix_waveletAgg	0.9 (0.77)	0.68 (0.77)
cluster_waveletLast	0.87 (0.73)	0.75 (0.73)

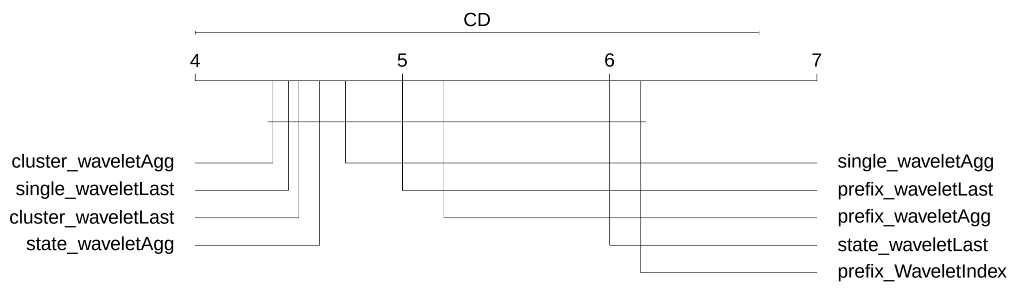
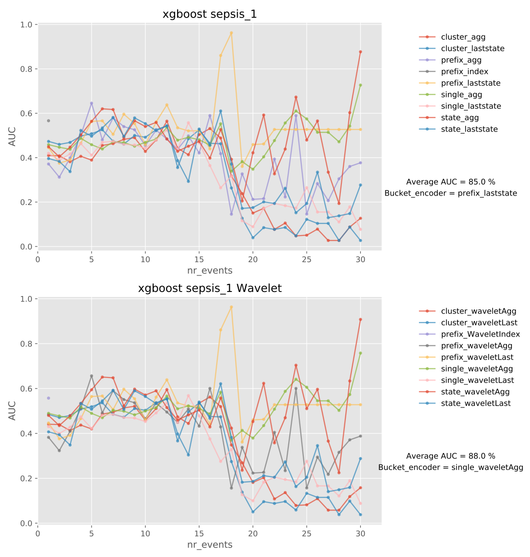


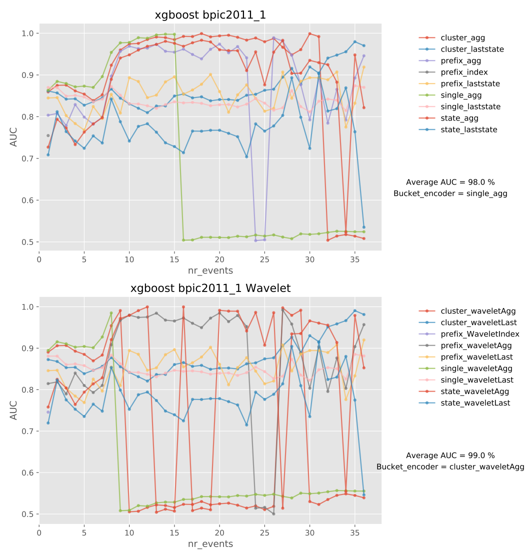
Figure 5.11: Critical difference diagram between all bucketing and encoding methods based on the highest AUC values using the Nemenyi test shows there is no statistical difference (at $pvalue < \alpha$) between all methods.

To compare our proposed complex sequence encoding method using discrete wavelet transformation and autoencoder, we used all classifiers with all previous encoding methods as a baseline; then we applied wavelet encoder with all classifiers. Figures 5.12, 5.13, and 5.14 show the results from the XGBoost classifier before and after applying wavelet encoder, and each subfigure contains the results for a specific event log in addition to the average AUC that associated with the best bucketing and encoding method for the baseline and the proposed method. We conclude from these experiments that using wavelet encoder with other simple encoding methods improved the overall AUC in most of the event logs, however for CatBoost the situation is different since the accuracy is not improved and this could happen because CatBoost classifier works very well with lots of categorical features and using wavelet encoder we decreased the available number of categorical attributes by adding more numeric features to the predictive models. Figures B.10 (CatBoost), B.13 (Logit), B.16 (RF), and B.19 (SVM) in the Appendix contain the comparison for all classifiers with and without the wavelet method.

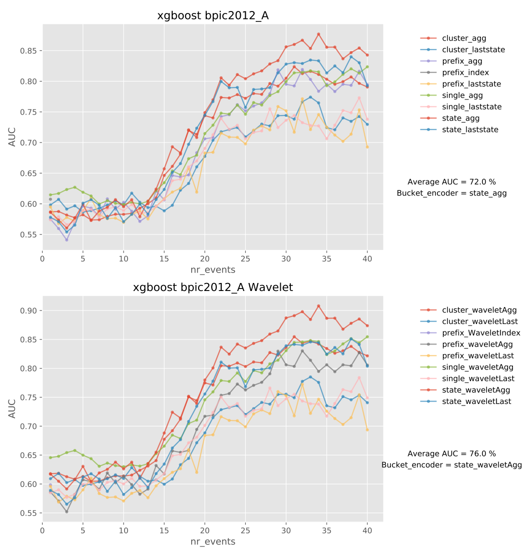


=

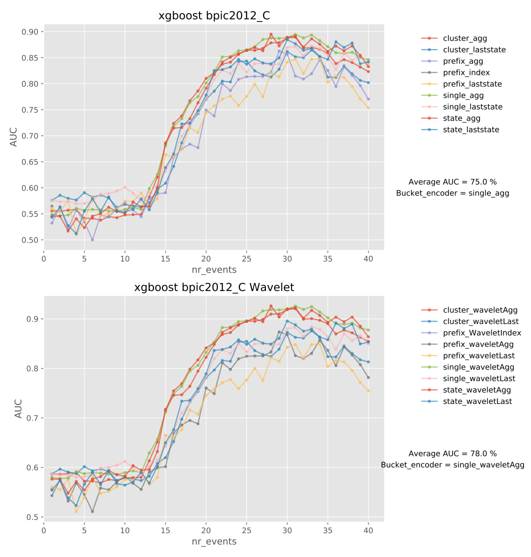
(a) Wavelet sepsis_1



(b) Wavelet bpic2011

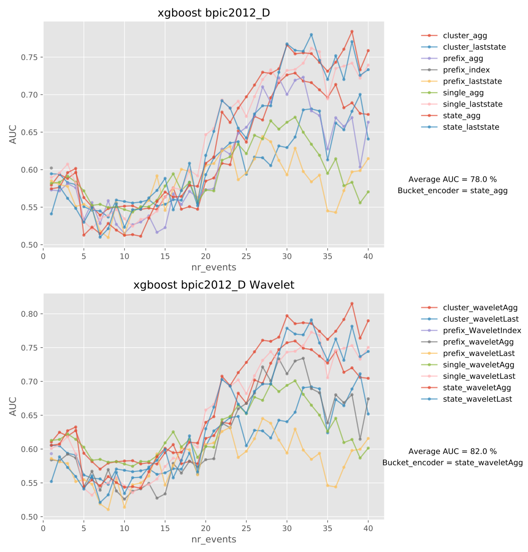


(c) Wavelet bpic2012_A

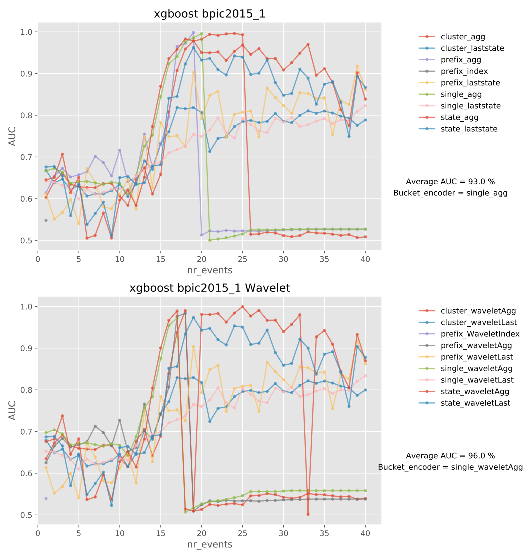


(d) Wavelet bpic2012_C

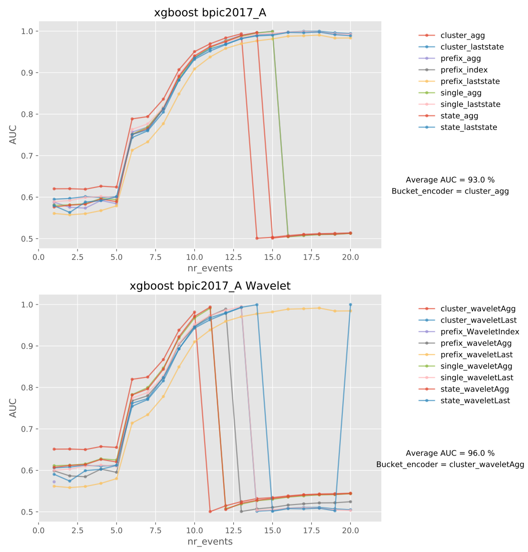
Figure 5.12: Comparing XGBoost before and after adding wavelet encoding on all event logs.

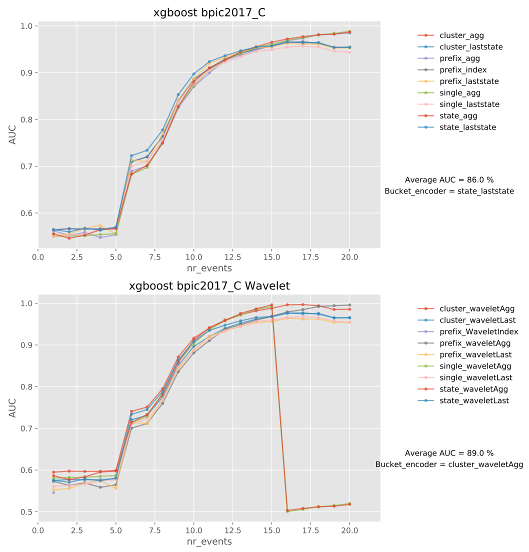


(a) Wavelet bpic2012_D



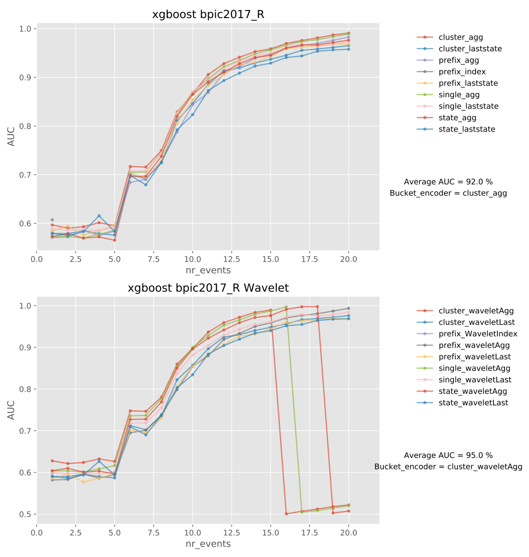
(b) Wavelet bpic2015



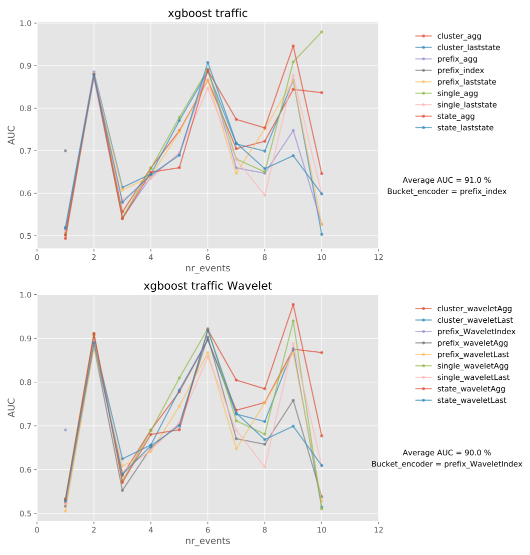


(c) Wavelet bpic2017_A **(d)** Wavelet bpic2017_C

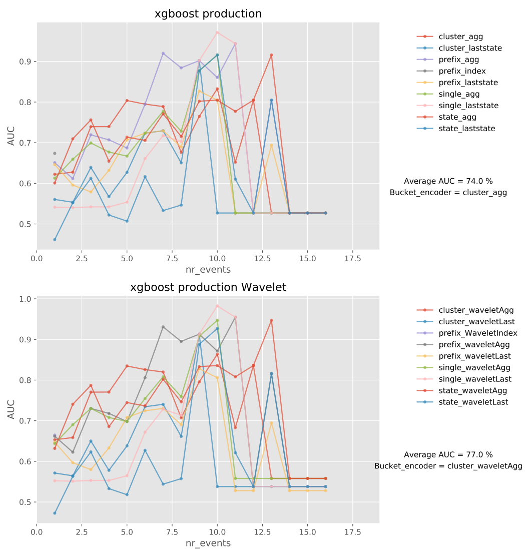
Figure 5.13: Comparing XGBoost before and after adding wavelet encoding on all event logs (*continued*).



(a) Wavelet bpic2017_R



(b) Wavelet traffic



(c) Wavelet Production

Figure 5.14: Comparing XGBoost before and after adding wavelet encoding on all event logs (*continued*)

5.3.3 Contribution 3: Inter-case features results

In this section, we added our proposed inter-case features that include three different categories, i.e. workload, demand intensity, and temporal context features, as shown in figure 5.15.

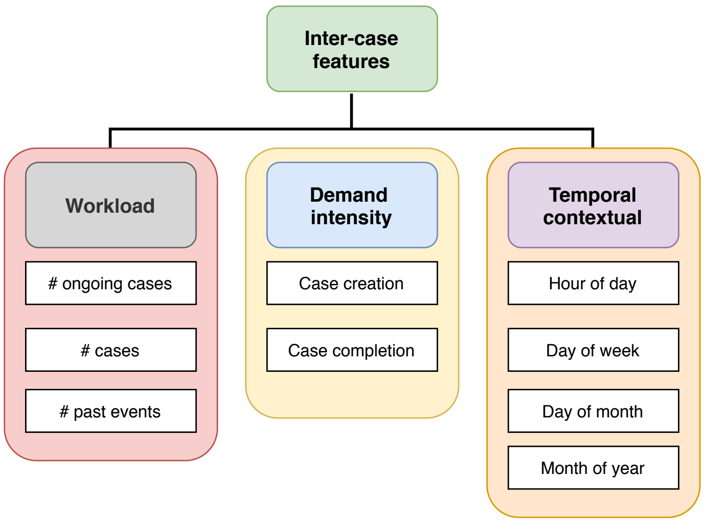


Figure 5.15: Categorization of inter-case features that we propose in this thesis, i.e. workload, demand intensity, and temporal context features.

To report the overall AUC after adding inter-case features, we used the same experiments that we did in figure 5.3. We calculated all prediction scores across all prefix lengths, and after that, we calculated the average score for all predictions based on the number of tested prefixes for each case as we did in our previously proposed methods.

Table 5.6 show collected results of CatBoost classifier with inter-case features in terms of AUC and F-score, and for XGBoost, RF, Logit, and SVM results are shown in Tables B.17, B.18, B.19, and B.20.

Results show that CatBoost has an improvement on 7 event logs in comparison to previously proposed methods (CatBoost, i.e. (10), and wavelet, i.e. (1)). Using CatBoost with inter-case features gives the best (or shared best) prediction score (i.e. AUC, and F-score) in 9 for AUC, and 12 for F-score out of 20 event logs. While for Xgboost, it achieves the best (or shared best) AUC in 10, and F-score in 6 event logs,

Table 5.6: Overall AUC (F-score) for **CatBoost plus Inter-case features**

	catboost					
	bpic2015_2	bpic2015_4	bpic2011_4	bpic2017_r	bpic2012_d	bpic2012_a
cluster_laststate	0.85 (0.23)	0.67 (0.35)	0.96 (0.87)	0.93 (0.69)	0.67 (0.3)	0.81 (0.81)
cluster_agg	0.99 (0.67)	0.7 (0.33)	0.96 (0.87)	0.93 (0.69)	0.7 (0.29)	0.82 (0.8)
single_agg	0.89 (0.61)	0.69 (0.33)	0.98 (0.94)	0.91 (0.65)	0.65 (0.25)	0.8 (0.81)
prefix_agg	0.94 (0.47)	0.67 (0.33)	**0.99** (**0.89**)	0.93 (0.67)	0.62 (0.2)	0.83 (0.83)
prefix_laststate	0.85 (0.31)	0.7 (0.26)	0.98 (0.89)	0.92 (0.68)	–	0.83 (0.83)
single_laststate	0.88 (0.39)	0.69 (0.26)	**0.99** (**0.93**)	0.92 (0.67)	–	0.82 (0.83)
prefix_index	0.73 (0.54)	0.72 (0.32)	**0.99** (**0.88**)	0.84 (0.42)	–	0.83 (0.7)
state_agg	0.85 (0.32)	**0.76** (**0.37**)	0.92 (0.84)	0.92 (0.67)	0.67 (0.29)	0.83 (0.82)
state_laststate	0.83 (0.33)	0.74 (0.33)	0.91 (0.81)	0.91 (0.65)	–	0.83 (0.81)

	catboost					
	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1	bpic2015_5
cluster_laststate	0.63 (0.27)	0.98 (0.93)	0.88 (0.83)	0.5 (0.0)	0.72 (0.38)	0.69 (0.45)
cluster_agg	0.6 (0.2)	0.99 (0.95)	0.98 (0.85)	0.5 (0.0)	0.84 (0.56)	0.69 (0.51)
single_agg	0.6 (0.22)	0.98 (0.93)	0.86 (0.79)	0.5 (0.0)	0.68 (0.34)	0.67 (0.49)
prefix_agg	0.61 (0.21)	0.98 (0.96)	0.84 (0.8)	0.5 (0.0)	0.72 (0.38)	0.71 (0.47)
prefix_laststate	0.6 (0.19)	0.99 (0.97)	0.83 (0.83)	0.5 (0.0)	0.71 (0.36)	0.69 (0.49)
single_laststate	0.6 (0.27)	0.98 (0.95)	0.88 (0.81)	0.5 (0.0)	0.86 (0.5)	0.71 (0.53)
prefix_index	0.6 (0.23)	0.99 (0.97)	0.87 (0.86)	**0.98** (**0.96**)	0.56 (0.15)	**0.72** (**0.36**)
state_agg	0.61 (0.19)	0.99 (0.94)	0.53 (0.74)	0.33 (0.0)	0.75 (0.44)	0.69 (0.5)
state_laststate	0.59 (0.32)	0.99 (0.9)	0.57 (0.75)	0.33 (0.0)	0.73 (0.46)	0.7 (0.55)

	catboost					
	sepsis_1	bpic2017_c	sepsis_3	bpic2012_c	production	bpic2011_1
cluster_laststate	0.46 (0.03)	**0.99** (**0.96**)	**0.98** (**0.9**)	0.97 (0.88)	0.69 (0.6)	0.99 (0.84)
cluster_agg	0.47 (0.05)	**0.99** (**0.97**)	**0.98** (**1.0**)	0.97 (0.87)	0.67 (0.62)	0.99 (0.87)
single_agg	0.52 (0.1)	**0.99** (**0.97**)	**0.98** (**0.91**)	0.97 (0.87)	0.67 (0.58)	0.97 (0.86)
prefix_agg	0.45 (0.01)	**0.99** (**0.97**)	**0.98** (**0.94**)	0.97 (0.87)	0.66 (0.6)	0.99 (0.93)
prefix_laststate	0.48 (0.02)	**0.99** (**0.97**)	**0.98** (**0.96**)	0.97 (0.87)	**0.78** (**0.55**)	0.99 (0.9)
single_laststate	0.55 (0.17)	**0.99** (**0.97**)	**0.98** (**0.94**)	0.97 (0.88)	0.77 (0.59)	0.97 (0.74)
prefix_index	0.51 (0.0)	**0.99** (**0.96**)	**0.98** (**1.0**)	0.97 (0.9)	0.75 (0.64)	**0.98** (**0.91**)
state_agg	0.45 (0.04)	**0.99** (**0.95**)	0.96 (0.92)	0.97 (0.87)	0.7 (0.59)	0.98 (0.9)
state_laststate	0.48 (0.11)	**0.99** (**0.97**)	0.96 (0.92)	0.97 (0.88)	0.74 (0.59)	0.98 (0.88)

	catboost
	bpic2017_a
cluster_laststate	**0.97** (**0.9**)
cluster_agg	0.96 (0.9)
single_agg	0.96 (0.89)
prefix_agg	0.96 (0.9)
prefix_laststate	0.96 (0.9)
single_laststate	**0.97** (**0.91**)
prefix_index	0.92 (0.84)
state_agg	0.96 (0.9)
state_laststate	0.96 (0.9)

SVM scores the best (or shared best) AUC in 1 event log. Logit, on average, does not give the equivalent level of scores as the other methods. On the other hand, for RF, it achieves the best (or shared best) AUC in 15, and F-score in 10 event logs, which is the best among all of them.

We used *A/B testing* to quantify the difference between all classifiers after adding our proposed inter-case features in terms of the performance. We employed the Nemenyi test as presented by ((47)) to plot the critical difference diagram as a statistical method to test whether the difference between all methods is significant or not with $\alpha = 0.05$. Critical difference diagram in figure 5.16 shows that the best performer classifier is RF with an average level of around 1.65; however, there is no statistical difference between it, XGBoost, and CatBoost.

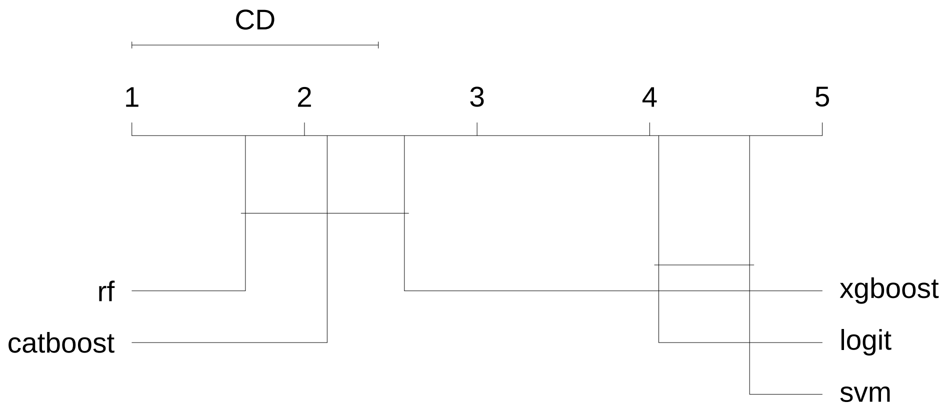


Figure 5.16: Critical difference diagram based on the highest AUC values using Nemenyi test shows there is no statistical difference (at $pvalue < \alpha$) between connected methods using a straight line such as CatBoost, XGBoost, and Random Forest.

For sequence encoding and trace bucketing methods, we can see in Table 5.7 different combinations between them and the number of event logs that each combination worked the best with CatBoost classifier. From these results, we conclude that the *prefix_index* and *cluster_agg* methods are the best across all combinations.

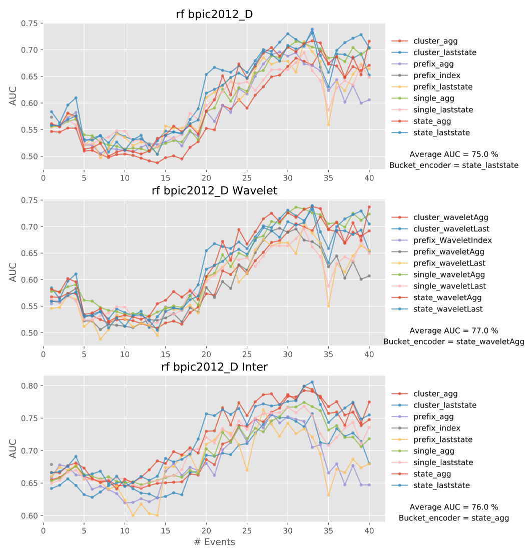
To compare our proposed inter-case features, we used all classifiers with all previous encoding methods as a baseline; then we added inter-case features to all event logs. Figures 5.17, 5.18, and 5.19, show the results from the RF classifier before and after

Table 5.7: Bucketing and sequence encoding methods after adding inter-case features.

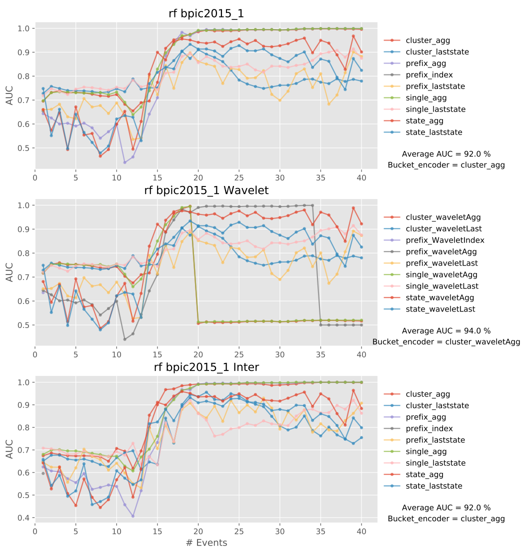
Bucket_Encoding_Method	# Data sets
cluster_agg_catboost	8
cluster_lastsatte_catboost	6
prefix_agg_catboost	6
prefix_index_catboost	9
prefix_laststate_catboost	6
single_agg_catboost	3
single_laststate_catboost	7
state_agg_catboost	5
state_laststate_catboost	4

applying inter-case features, and each subfigure contains the results for a specific event log in addition to the average AUC associated with the best bucketing and encoding method for the baseline and the proposed method.

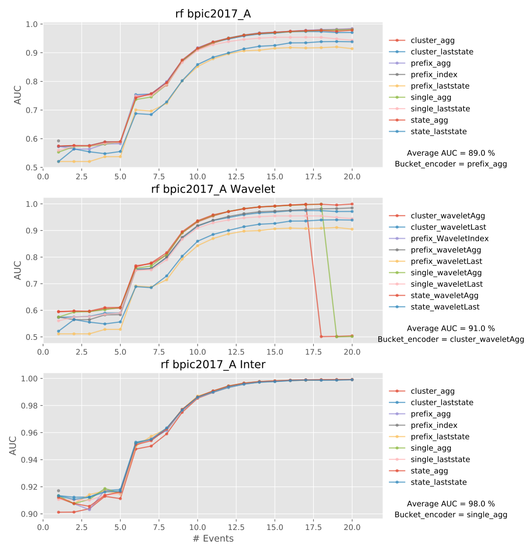
We conclude from inter-case features experiments that the overall prediction scores in terms of the AUC are improved on most of the event logs with all classification methods. Figures B.22 (CatBoost), B.25 (XGBoost), B.28 (Logit), and B.32 (SVM) in the Appendix show comparison between all classification, sequence encoding, and trace bucketing techniques before and after adding inter-case features.



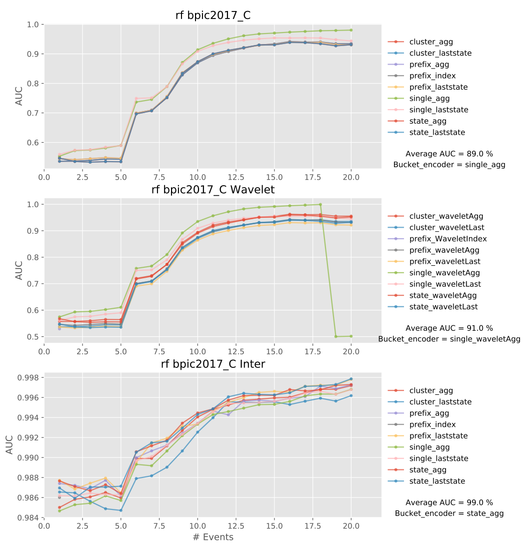
(a) Bpic2012_D



(b) Bpic2015

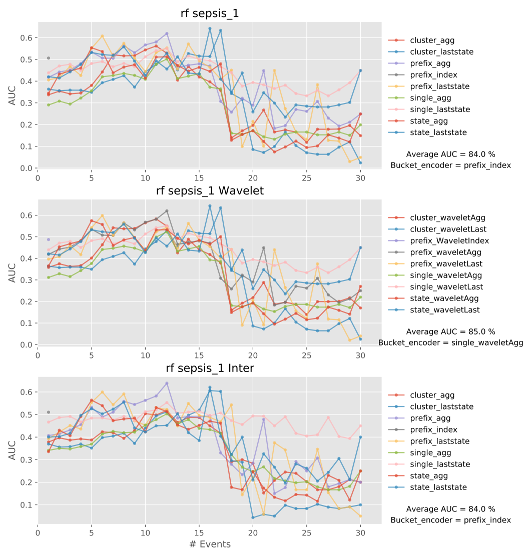


(c) Bpic2017_A

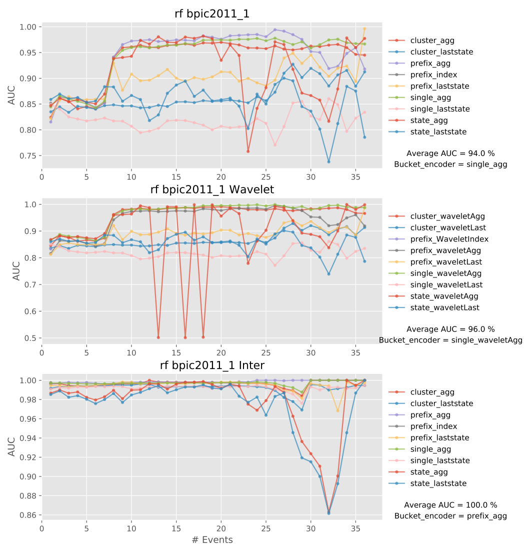


(d) Bpic2017_C

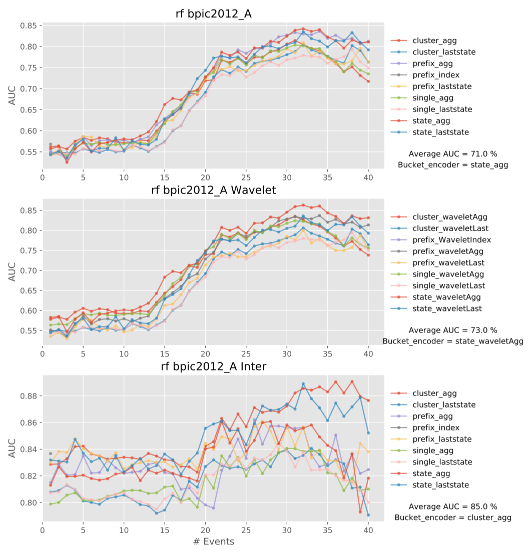
Figure 5.17: Comparing RF before and after adding Inter-case features on all event logs



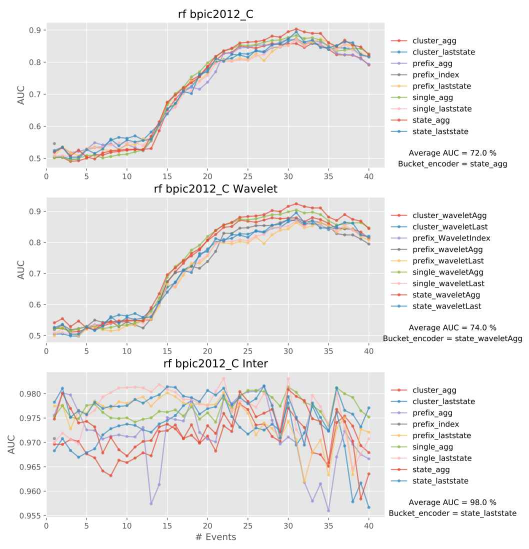
(a) Sepsis_1



(b) Bpic2011

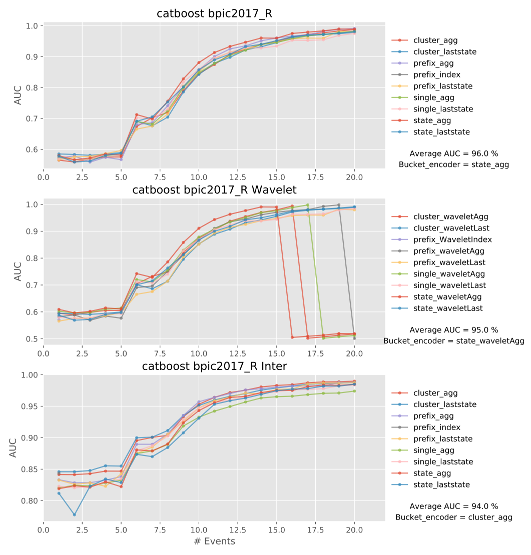


(c) Bpic2012_A



(d) Bpic2012_C

Figure 5.18: Comparing RF before and after adding Inter-case features on all event logs (*continued*).


(a) Wavelet bpic2017_R

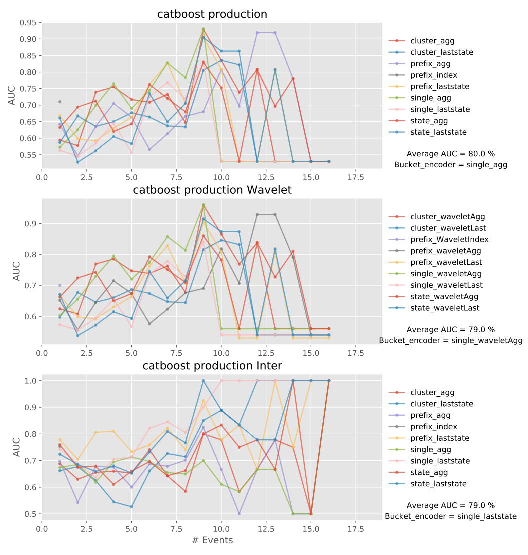

(b) Production

Figure 5.19: Comparing RF before and after adding Inter-case features on all event logs (*continued*)

5.3.3.1 Inter-case features interpretability

After adding inter-case features to our predictive models, we achieved a very good performance in terms of accuracy, which leads us to analyze and understand how and why our model achieved this jump in performance. To understand the behaviour of our outcome-oriented predictive models after adding inter-case features, we used a consolidated framework (i.e. *SHAP*) to explain how predictions are made using these models. ***SH**apley **A**dditive ex**P**lanations* gives every feature a weight that refers to the importance of this feature for a specific prediction (48). Particularly, SHAP values attempt to describe the outcome of a model as a total amount of the effects of every feature being entered into a conditional expectation (49).

In this thesis, we worked with binary classification problems, which means our target label has two possible values 0 or 1 (or $\{0, 1\}$). SHAP provides us with a detailed logic about which inter-case features have the highest effect or impact in the model. We used *Production* event log with SHAP analysis. For example, considering the right decision of predicting the number of discarded product orders is less than zero, which represents our *false* or zero outcomes. Figure 5.20 shows an explanation of features where every one of them contributing to shifting the model outcome from the goal value (the common model outcome across the training event log we passed) to the original model outcome. Inter-case features forcing the forecast higher are displayed in *red* such as *open_cases*, and those forcing the forecast lower are in *blue* as *nr_ongoing_cases*. This figure is related to only one case from the event log, and the outcome for it is negative or false. On the other hand, for another case from the event log with an actual positive outcome, we can see from figure 5.21 different inter-case features that force the predictive model towards a specific output.

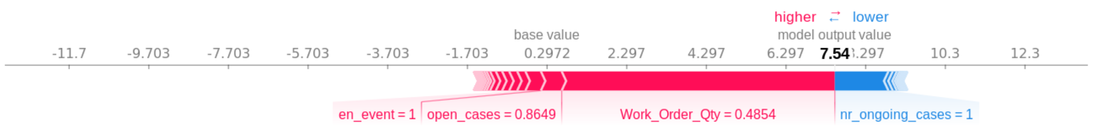


Figure 5.20: SHAP value analysis explanation when predicting the number of discarded product orders is less than zero. Features with red colour are pushing model higher for prediction and vice verse with blue.

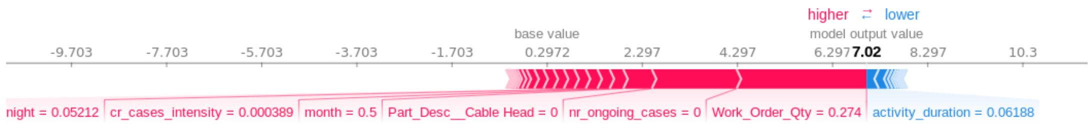


Figure 5.21: SHAP value analysis explanation when predicting the number of discarded product orders is more than zero. Features with red colour are pushing model higher for prediction and vice verse with blue.

Figures 5.20, and 5.21 both contain an inter-case feature, i.e. *(nr_ongoing_cases)*, that we introduced in this thesis which captures information about the workload of business processes (see section 4.3), and this feature is helping and pushing the model toward the correct prediction either for false or negative outcome correctly, so we decided to explore and interpret this inter-case feature more.

Figure 5.22 shows the effect of using *nr_ongoing_cases* on different cases form our event log, i.e. around 700 samples. Results show the nr_ongoing_cases feature has a high impact on the model outcome either for false or positive cases, as shown in blue and red, respectively.

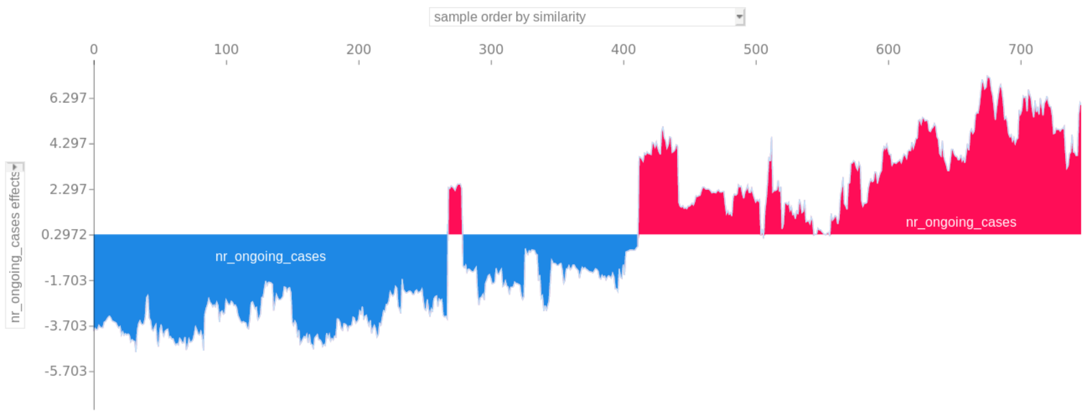
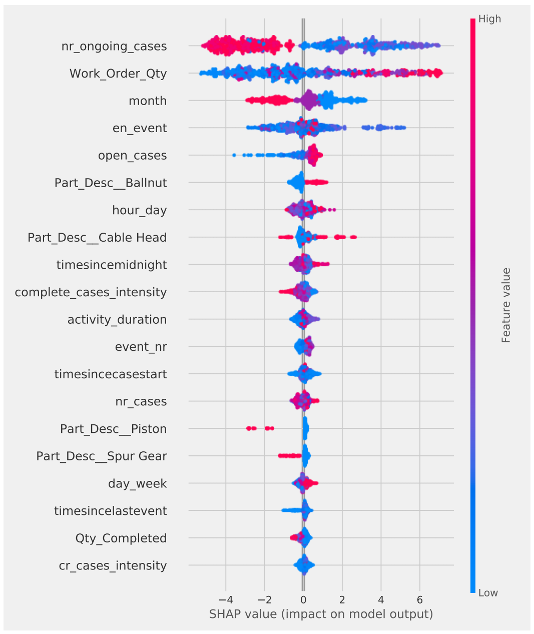


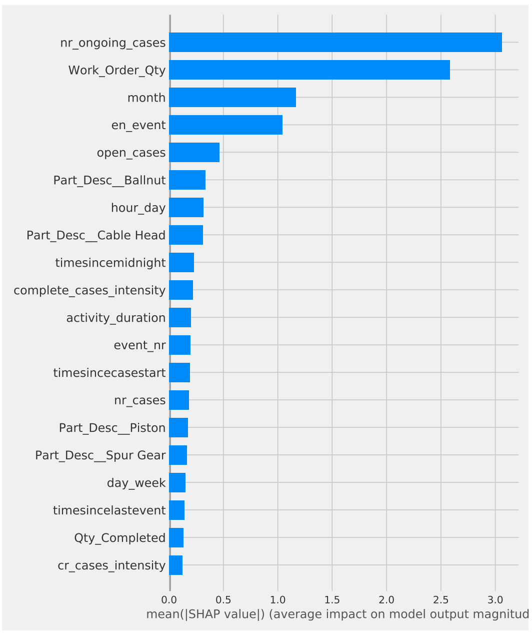
Figure 5.22: SHAP value analysis explanation when predicting the number of discarded product orders is more or less than zero in different cases. Features with red colour are pushing model higher for prediction and vice verse with blue.

Figure 5.23a shows a traditional bar chart on the basis of the average of the SHAP value measures over the event log and plots it as a simple bar chart. While, figure 5.23b, shows a density scatter plot of SHAP values for every feature to distinguish how much influence every feature has on the model outcome for cases in the validation set. In both figures 5.23, features are sorted on the basis of the sum of SHAP value measures across overall cases. It is intriguing to remark that one of our introduced inter-case feature that is *nr_ongoing_cases* has a huge impact on the model outcome, and it has the highest SHAP value among all other features. Also, in both figures, we can see three other inter-case features *(i.e. hour_day, nr_cases, and cr_cases_intensity)* that have a significant SHAP value among all other features in the original event log that reflects on the model output.

Figure 5.23: SHAP density plot of



(a) SHAP density plot of SHAP values for every feature to recognize how serious impact every feature holds on the model output



(b) SHAP feature importance plot based on the average of the SHAP value measures over the event log

Finally, we can see in figure 5.24, a SHAP dependence plot that shows the impact of a one (or pair) feature over the entire event log. Partial dependence plot (PDP) shows a relation between feature values and SHAP values which is corresponding to this same feature over different cases. Exactly the same as we saw before a higher number of ongoing cases have a significant impact on the positive outcome of the model, and vice versa for a negative outcome.

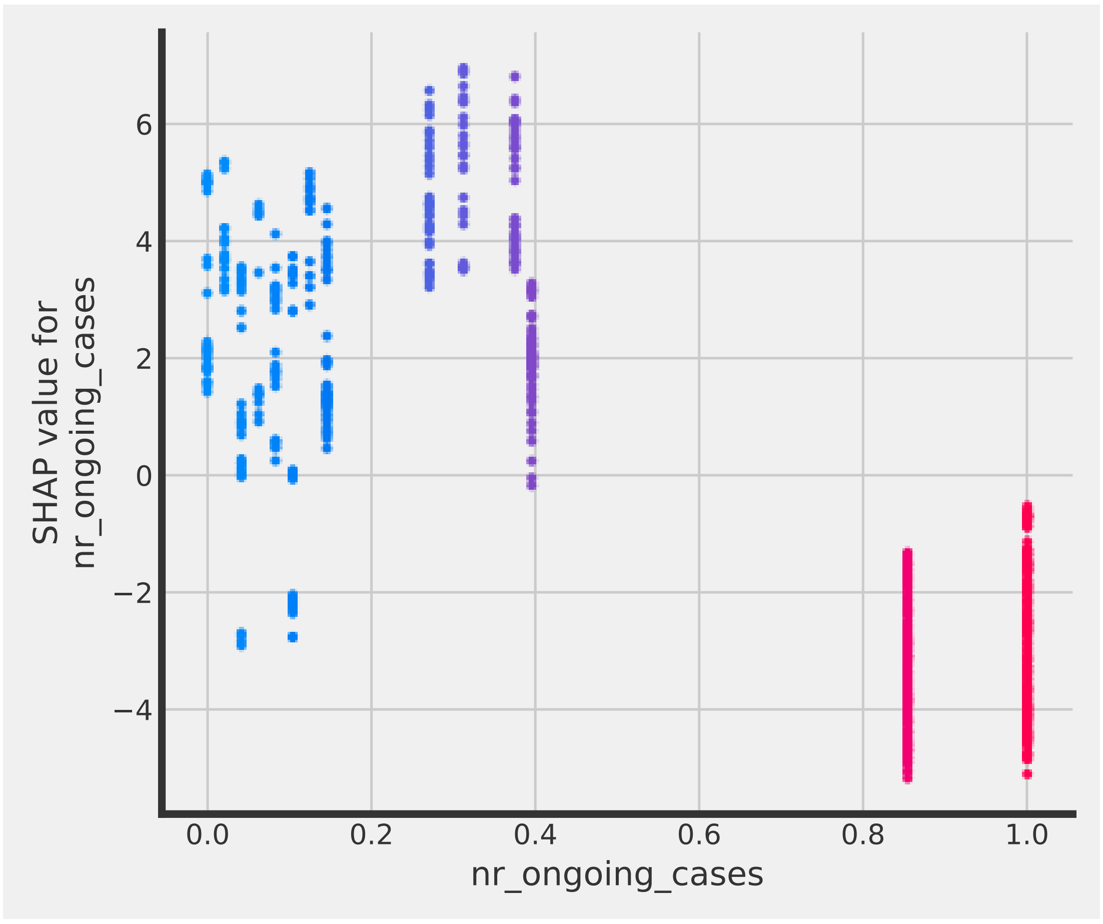


Figure 5.24: SHAP dependence plots

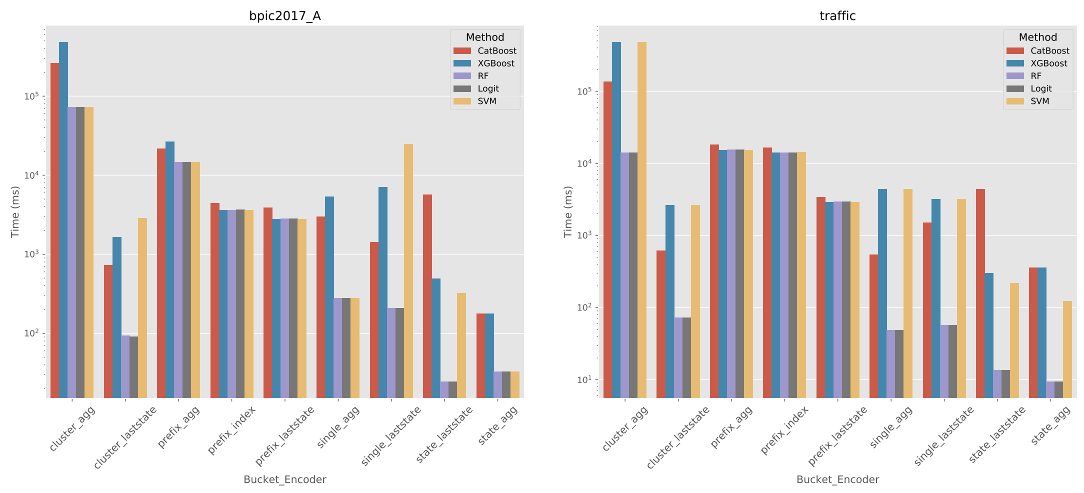
After interpreting our outcome-oriented models based on the proposed inter-case features using SHAP analysis, we conclude that four out of nine proposed inter-case features have a significant impact on the model outcome. Accordingly, the jump in performance values for different classification models happen due to the huge effect of our inter-case features in particular *nr_ongoing_cases* feature.

5.4 Results: time performance

In this section, we report the time measurements which is corresponding to the efficiency metric (see section 5.2.3) for all methods that we proposed in this thesis.

5.4.1 CatBoost results:

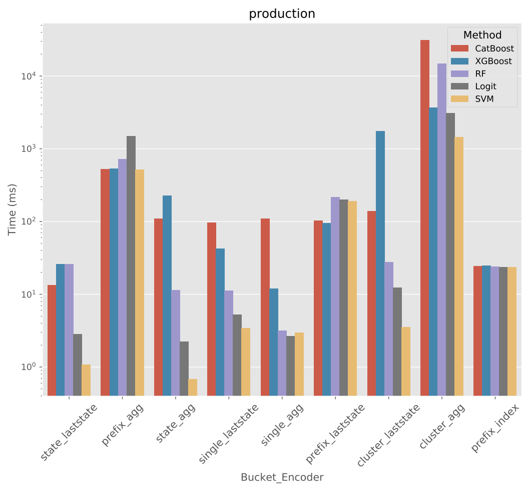
Predictive models are assumed to take much time during the training process while in the prediction time, it is better to be near real-time. In offline (or training) phase using CatBoost with larger event log such as *BPIC2017, or Traffic fine* (see figure 5.25) on average is faster than XGBoost. On the other hand, using it with smaller event logs such as *Production* it's more or less the same as XGBoost (see figure 5.26). *Sparse* data is a bottleneck to CatBoost during the training step, and it becomes slower than XGBoost and other methods. In online (or prediction) time we can say that CatBoost is on average the fastest method (see figure 5.27) in most of the event logs and it does not matter if the original event log is bigger or not because we are dealing with a stream of incoming events.



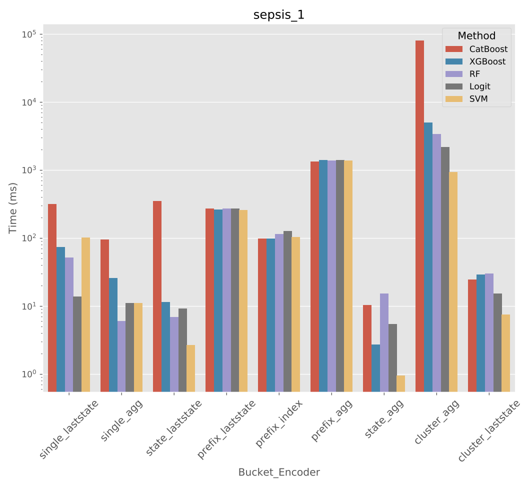
(a) Execution time bpic2017_A **(b)** Execution time Traffic

Figure 5.25: Execution time for models on large event logs such as BPIC2017, and Traffic fine logs. (training time)

Execution times for all bucketing, encoding and classification methods are computed across five equal executions employing the best optimal parameters, are shown in Table

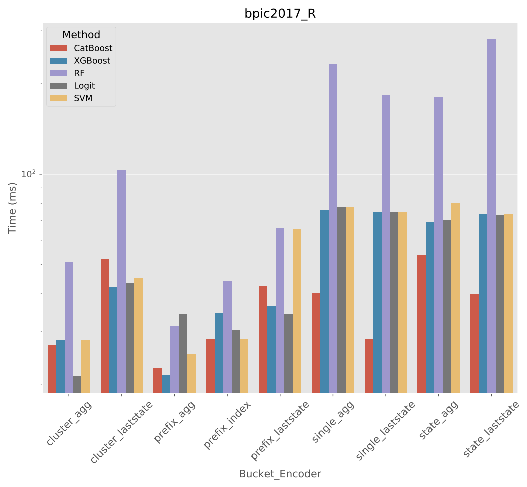


(a) Execution time Production

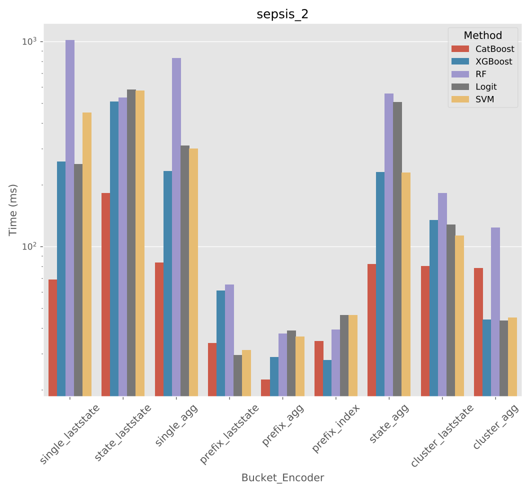


(b) Execution time Sepsis_1

Figure 5.26: Execution time for models on small event logs such as Production, and Sepsis cases logs. (training time)



(a) Execution time BPIC2017_R (online)



(b) Execution time Sepsis_2 (online)

Figure 5.27: Execution time for models on large and small event logs such as BPIC2017, and Sepsis logs. (prediction time)

Table 5.8: Execution times for **CatBoost**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	4035.1 ± 27.6	77.51 ± 0.05	5392.83 ± 42.3	55.1 ± 0.03	**67.48 ± 43.54**	115.84 ± 0.06
cluster_agg	425067.1 ± 24.19	24.84 ± 0.03	112428.33 ± 26.27	29.36 ± 0.03	20969.05 ± 25.85	26.38 ± 0.03
single_agg	407.26 ± 45.59	82.14 ± 0.03	297.85 ± 41.59	28.26 ± 0.01	75.38 ± 26.51	156.81 ± 0.06
prefix_agg	8788.2 ± 22.76	38.55 ± 40.05	8829.17 ± 31.6	37.0 ± 30.16	8260.38 ± 30.58	36.17 ± 22.18
prefix_laststate	1575.81 ± 24.55	28.33 ± 44.55	2087.29 ± 35.0	39.28 ± 34.76	1084.4 ± 41.17	60.54 ± 37.11
single_laststate	**255.6 ± 28.38**	60.07 ± 0.02	525.63 ± 45.56	**27.49 ± 0.01**	111.92 ± 41.15	44.08 ± 0.01
prefix_index	366.63 ± 23.15	23.34 ± 34.86	874.61 ± 26.28	30.16 ± 31.67	208.21 ± 25.37	44.1 ± 32.57
state_agg	276.57 ± 44.63	**18.43 ± 0.01**	3130.88 ± 30.61	42.69 ± 0.01	1742.07 ± 32.16	26.05 ± 0.01
state_laststate	337.11 ± 40.79	31.68 ± 0.01	**20.36 ± 40.39**	32.51 ± 0.01	257.69 ± 21.02	**20.94 ± 0.0**

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	815.78 ± 35.58	48.4 ± 0.02	732.62 ± 32.4	65.62 ± 0.02	340.59 ± 26.76	103.05 ± 0.05
cluster_agg	214560.64 ± 23.54	**26.48 ± 0.03**	262684.18 ± 24.33	34.85 ± 0.03	136208.45 ± 38.74	61.2 ± 0.06
single_agg	1456.56 ± 40.4	28.45 ± 0.01	2985.75 ± 21.3	43.63 ± 0.01	**84.08 ± 31.63**	44.92 ± 0.02
prefix_agg	8531.68 ± 24.16	32.01 ± 31.66	21626.27 ± 43.03	**18.77 ± 23.1**	3733.45 ± 22.92	28.76 ± 34.18
prefix_laststate	1623.02 ± 38.17	37.97 ± 36.06	3918.62 ± 24.78	29.37 ± 32.87	679.86 ± 38.96	54.3 ± 48.75
single_laststate	**688.33 ± 47.94**	27.68 ± 0.01	1424.59 ± 32.63	26.28 ± 0.01	147.98 ± 42.73	**16.53 ± 0.0**
prefix_index	1157.14 ± 31.24	40.83 ± 39.4	4424.61 ± 32.41	24.95 ± 44.41	122.02 ± 38.28	23.27 ± 43.07
state_agg	4053.21 ± 22.2	42.79 ± 0.01	**178.05 ± 29.37**	69.55 ± 0.02	317.15 ± 46.9	59.55 ± 0.02
state_laststate	18068.63 ± 33.76	33.61 ± 0.01	5698.17 ± 28.83	41.78 ± 0.01	280.86 ± 42.4	98.96 ± 0.03

	traffic		**bpic2012_A**		**bpic2015_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	617.97 ± 39.02	191.29 ± 0.03	5726.52 ± 28.41	53.35 ± 0.03	**141.09 ± 23.81**	89.93 ± 0.05
cluster_agg	136871.63 ± 24.19	105.33 ± 0.04	292502.98 ± 45.11	25.63 ± 0.03	146380.45 ± 49.55	27.19 ± 0.03
single_agg	545.66 ± 39.0	132.31 ± 0.02	**167.15 ± 45.24**	**22.56 ± 0.01**	251.5 ± 49.6	178.36 ± 0.07
prefix_agg	18164.8 ± 36.76	36.3 ± 25.2	7348.92 ± 45.24	34.7 ± 38.34	17847.98 ± 42.98	34.85 ± 32.02
prefix_laststate	3421.21 ± 44.07	55.12 ± 38.89	1604.91 ± 40.15	63.84 ± 32.87	2360.49 ± 25.56	57.44 ± 44.94
single_laststate	1508.39 ± 49.12	85.48 ± 0.01	502.66 ± 49.0	27.52 ± 0.01	307.66 ± 27.58	43.78 ± 0.01
prefix_index	16493.67 ± 45.46	**24.27 ± 33.08**	1162.18 ± 48.76	24.65 ± 29.64	519.29 ± 31.66	41.86 ± 44.17
state_agg	**355.86 ± 43.55**	228.84 ± 0.03	752.18 ± 21.48	75.63 ± 0.02	5062.38 ± 31.58	**19.79 ± 0.01**
state_laststate	4411.24 ± 23.22	126.22 ± 0.02	1227.07 ± 44.28	45.05 ± 0.01	348.02 ± 42.56	29.17 ± 0.01

	bpic2011_3		**bpic2011_2**		**sepsis_2**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	**89.67 ± 48.8**	113.89 ± 0.05	**49.63 ± 22.59**	105.55 ± 0.05	95.29 ± 32.56	80.4 ± 0.03
cluster_agg	86472.7 ± 44.97	107.02 ± 0.1	61273.31 ± 32.75	95.88 ± 0.11	2862494.55 ± 41.88	78.42 ± 0.06
single_agg	154.05 ± 28.92	58.65 ± 0.02	103.01 ± 34.63	49.87 ± 0.02	**50.57 ± 46.54**	83.61 ± 0.02
prefix_agg	1640.77 ± 45.98	38.4 ± 38.76	3522.71 ± 45.65	21.45 ± 35.65	475.74 ± 40.06	**22.46 ± 37.92**
prefix_laststate	327.68 ± 25.27	39.2 ± 48.2	762.27 ± 47.93	31.09 ± 34.11	90.68 ± 23.82	33.84 ± 21.77
single_laststate	101.73 ± 39.45	**20.95 ± 0.01**	202.65 ± 33.1	**15.78 ± 0.0**	2061.4 ± 23.47	69.08 ± 0.01
prefix_index	114.68 ± 28.55	48.19 ± 45.26	127.64 ± 28.45	41.89 ± 31.75	103.84 ± 40.16	34.69 ± 26.72
state_agg	422.62 ± 29.68	82.29 ± 0.02	523.02 ± 33.51	59.7 ± 0.02	3822.72 ± 25.15	82.25 ± 0.02
state_laststate	343.87 ± 43.47	98.45 ± 0.02	193.67 ± 27.01	90.83 ± 0.03	91.67 ± 29.95	182.92 ± 0.03

	bpic2015_1		**bpic2015_5**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	92.35 ± 47.68	49.43 ± 0.02	165.75 ± 21.57	110.39 ± 0.06	2046.66 ± 37.68	64.78 ± 0.02
cluster_agg	2614134.85 ± 45.94	30.22 ± 0.03	19001.6 ± 39.57	33.35 ± 0.04	16298503.69 ± 45.75	57.33 ± 0.05
single_agg	185.25 ± 35.31	77.48 ± 0.03	122.79 ± 30.3	212.25 ± 0.08	1255.99 ± 25.72	72.8 ± 0.02
prefix_agg	7671.38 ± 31.1	34.54 ± 42.26	11688.96 ± 34.73	35.25 ± 35.74	1402.76 ± 43.85	**31.66 ± 27.51**
prefix_laststate	1358.18 ± 44.4	45.65 ± 20.76	2022.84 ± 34.34	32.79 ± 48.92	270.83 ± 22.22	48.13 ± 34.91
single_laststate	80.92 ± 45.55	65.91 ± 0.02	610.51 ± 20.8	59.06 ± 0.02	126.26 ± 35.17	57.55 ± 0.01
prefix_index	256.4 ± 21.77	48.89 ± 34.32	521.86 ± 38.13	43.87 ± 32.48	105.52 ± 31.67	38.86 ± 29.15
state_agg	**64.95 ± 35.87**	19.7 ± 0.01	147.38 ± 42.92	24.11 ± 0.01	4657.35 ± 27.14	66.51 ± 0.02
state_laststate	120.37 ± 49.72	**16.41 ± 0.0**	**100.01 ± 44.41**	**14.73 ± 0.0**	**32.92 ± 38.47**	143.57 ± 0.03

	bpic2017_C		**production**		**sepsis_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	10772.93 ± 33.26	68.79 ± 0.03	139.26 ± 44.45	64.07 ± 0.02	24.53 ± 39.11	61.16 ± 0.02
cluster_agg	114443.62 ± 28.67	36.6 ± 0.03	31414.7 ± 43.16	43.86 ± 0.03	80600.81 ± 33.14	70.65 ± 0.06
single_agg	2270.8 ± 35.67	40.6 ± 0.01	110.04 ± 38.96	127.63 ± 0.03	96.49 ± 44.2	73.18 ± 0.02
prefix_agg	20727.33 ± 36.81	40.96 ± 46.28	530.54 ± 36.03	**20.01 ± 39.1**	1348.64 ± 46.77	**26.36 ± 28.4**
prefix_laststate	2816.38 ± 31.59	32.16 ± 39.09	103.66 ± 45.82	28.38 ± 47.32	271.41 ± 45.06	40.03 ± 47.92
single_laststate	541.76 ± 25.16	29.1 ± 0.01	96.73 ± 36.41	32.89 ± 0.0	320.38 ± 32.87	62.25 ± 0.01
prefix_index	4205.79 ± 27.01	**22.88 ± 49.9**	24.45 ± 26.57	32.09 ± 41.17	99.68 ± 28.32	26.85 ± 35.73
state_agg	756.35 ± 34.38	59.65 ± 0.02	110.15 ± 21.7	101.65 ± 0.02	**10.5 ± 21.92**	46.98 ± 0.01
state_laststate	**246.39 ± 33.0**	74.08 ± 0.02	**13.4 ± 44.45**	48.59 ± 0.01	351.73 ± 23.16	127.0 ± 0.02

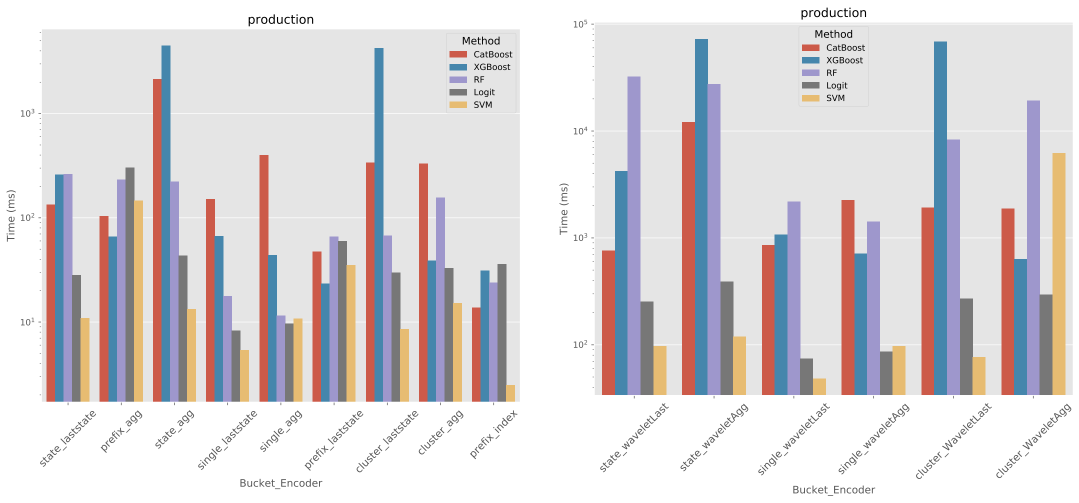
	bpic2011_1		**bpic2017_R**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	848.63 ± 20.93	104.77 ± 0.05	837.23 ± 21.12	52.23 ± 0.02
cluster_agg	65724.68 ± 47.54	65.95 ± 0.05	7436286.17 ± 24.4	27.04 ± 0.02
single_agg	**34.98 ± 42.04**	57.16 ± 0.03	8619.04 ± 44.48	40.29 ± 0.01
prefix_agg	2262.18 ± 42.89	25.48 ± 36.77	23969.9 ± 25.89	**22.68 ± 21.44**
prefix_laststate	485.91 ± 21.47	46.61 ± 30.32	4084.31 ± 40.61	42.28 ± 40.28
single_laststate	68.88 ± 32.63	**18.22 ± 0.0**	5017.23 ± 39.54	28.33 ± 0.01
prefix_index	115.67 ± 40.37	49.92 ± 24.23	4604.51 ± 24.8	28.19 ± 43.45
state_agg	1637.66 ± 42.2	74.57 ± 0.02	8506.42 ± 28.53	53.61 ± 0.01
state_laststate	469.56 ± 44.77	100.15 ± 0.03	**344.58 ± 28.21**	39.75 ± 0.01

5.8 (CatBoost), Table B.5 (XGBoost), Table B.6 (Logit), Table B.7 (RF), and Table B.8 (SVM).

5.4.2 Complex sequence encoding (Wavelet) results:

Wavelet encoding method achieved improvement w.r.t the prediction accuracy in most of the event logs, however, for the execution time it takes much time just to encode each event log into a feature matrix as the time complexity for this method is $O(n^3)$. Table 5.9 shows the execution time of wavelet encoding and CatBoost classifier. Table B.13 (XGBoost), Table B.14 (RF), Table B.15 (Logit), and Table B.16 (SVM).

To show the difference between offline and online execution times, we compare this execution time with the previous encoding methods in contribution 1 section 5.4.1 results. For example, figure 5.28 shows the difference between wavelet and other techniques w.r.t production event log during the training process, while figure 5.29 shows the same compression but for prediction (or online) phase. We can conclude that using the wavelet encoding method; our predictive model suffers from a huge amount of time to train and to predict any running business case, which could be a good direction in future work to improve the time complexity of this encoding method.



(a) Without wavelet

(b) With wavelet

Figure 5.28: Execution time before and after adding the wavelet encoding method (training) on Production event log

Table 5.9: Execution times for **Wavelet plus CatBoost**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	22879.0 ± 156.51	439.49 ± 0.28	30577.36 ± 239.83	312.39 ± 0.16	**382.61 ± 246.88**	656.8 ± 0.36
prefix_waveletLast	8934.83 ± 139.22	160.65 ± 252.58	11834.96 ± 198.46	222.73 ± 197.11	6148.53 ± 233.44	343.24 ± 210.43
state_waveletAgg	1568.13 ± 253.02	**104.51 ± 0.03**	17752.08 ± 173.56	242.03 ± 0.07	9877.56 ± 182.33	147.71 ± 0.05
single_waveletLast	**1449.25 ± 160.91**	340.61 ± 0.1	2980.32 ± 258.32	**155.86 ± 0.04**	634.6 ± 233.32	249.95 ± 0.08
single_waveletAgg	2309.17 ± 258.5	465.72 ± 0.18	1688.82 ± 235.79	160.24 ± 0.06	427.41 ± 150.31	889.14 ± 0.33
state_waveletLast	1911.44 ± 231.25	179.62 ± 0.06	**115.45 ± 229.03**	184.33 ± 0.04	1461.12 ± 119.17	**118.75 ± 0.03**
prefix_waveletAgg	49829.08 ± 129.06	218.57 ± 227.11	50061.39 ± 179.17	209.77 ± 170.98	46836.34 ± 173.41	205.06 ± 125.74
cluster_WaveletAgg	2410130.43 ± 137.17	140.82 ± 0.15	637468.64 ± 148.93	166.49 ± 0.18	118894.5 ± 146.59	149.58 ± 0.15
prefix_waveletIndex	2078.79 ± 131.29	132.36 ± 197.68	4959.04 ± 149.02	170.99 ± 179.56	1180.58 ± 143.87	250.05 ± 184.69

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	4625.45 ± 201.71	274.44 ± 0.13	4153.96 ± 183.72	372.06 ± 0.14	1931.14 ± 151.72	584.32 ± 0.3
prefix_waveletLast	9202.54 ± 216.43	215.31 ± 204.45	22218.56 ± 140.5	166.51 ± 186.38	3854.81 ± 220.92	307.87 ± 276.41
state_waveletAgg	22981.67 ± 125.89	242.6 ± 0.08	**1009.56 ± 166.52**	394.38 ± 0.09	1798.21 ± 265.92	337.64 ± 0.1
single_waveletLast	**3902.82 ± 271.8**	156.92 ± 0.04	8077.42 ± 184.99	149.0 ± 0.03	839.07 ± 242.28	**93.72 ± 0.03**
single_waveletAgg	8258.69 ± 229.09	161.29 ± 0.07	16929.19 ± 120.77	247.36 ± 0.07	**476.71 ± 179.34**	254.71 ± 0.11
state_waveletLast	102449.13 ± 191.44	190.57 ± 0.06	32308.61 ± 163.49	236.88 ± 0.05	1592.47 ± 240.39	561.09 ± 0.15
prefix_waveletAgg	48374.64 ± 137.01	181.48 ± 179.54	122620.94 ± 243.97	**106.42 ± 131.0**	21168.69 ± 129.97	163.08 ± 193.8
cluster_WaveletAgg	1216558.85 ± 133.45	**150.15 ± 0.15**	1489419.31 ± 137.93	197.6 ± 0.16	772301.89 ± 219.67	347.0 ± 0.32
prefix_waveletIndex	6560.96 ± 177.14	231.5 ± 223.4	25087.55 ± 183.79	141.49 ± 251.78	691.86 ± 217.05	131.92 ± 244.23

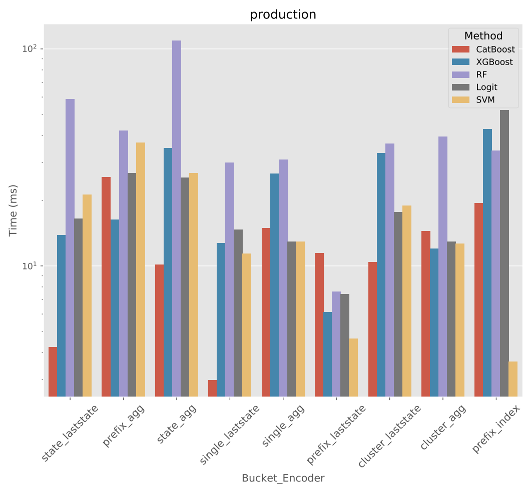
	traffic		**bpic2012_A**		**bpic2015_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	3503.88 ± 221.25	1084.64 ± 0.18	32469.38 ± 161.08	302.52 ± 0.15	**799.96 ± 134.97**	509.89 ± 0.27
prefix_waveletLast	19398.28 ± 249.89	312.52 ± 220.48	9099.81 ± 227.66	361.97 ± 186.37	13383.99 ± 144.93	325.68 ± 254.79
state_waveletAgg	**2017.72 ± 246.95**	1297.52 ± 0.14	4264.84 ± 121.79	428.82 ± 0.14	28703.68 ± 179.04	**112.22 ± 0.03**
single_waveletLast	8552.55 ± 278.53	484.7 ± 0.05	2850.08 ± 277.83	156.02 ± 0.04	1744.44 ± 156.36	248.21 ± 0.08
single_waveletAgg	3093.89 ± 221.15	750.19 ± 0.1	**947.75 ± 256.5**	**127.93 ± 0.05**	1425.98 ± 281.21	1011.33 ± 0.39
state_waveletLast	25011.73 ± 131.64	715.64 ± 0.09	6957.5 ± 251.05	255.41 ± 0.07	1973.28 ± 241.29	165.37 ± 0.05
prefix_waveletAgg	102994.44 ± 208.44	205.82 ± 142.91	41668.39 ± 256.53	196.72 ± 217.41	101198.07 ± 243.7	197.61 ± 181.53
cluster_WaveletAgg	776062.15 ± 137.14	597.23 ± 0.22	1658491.89 ± 255.75	145.35 ± 0.15	829977.17 ± 280.96	154.18 ± 0.15
prefix_waveletIndex	93519.12 ± 257.73	**137.62 ± 187.55**	6589.57 ± 276.5	139.76 ± 168.08	2944.38 ± 179.5	237.32 ± 250.45

	bpic2011_3		**bpic2011_2**		**sepsis_2**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	**508.42 ± 276.7**	645.74 ± 0.31	**281.39 ± 128.11**	598.44 ± 0.31	540.32 ± 184.63	455.86 ± 0.15
prefix_waveletLast	1857.93 ± 143.26	222.26 ± 273.29	4322.05 ± 271.76	176.27 ± 193.39	514.16 ± 135.06	191.86 ± 123.44
state_waveletAgg	2396.27 ± 168.3	466.6 ± 0.13	2965.53 ± 189.97	338.51 ± 0.11	21674.84 ± 142.6	466.33 ± 0.09
single_waveletLast	576.83 ± 223.69	**118.76 ± 0.03**	1149.02 ± 187.66	**89.48 ± 0.03**	11688.12 ± 133.08	391.66 ± 0.08
single_waveletAgg	873.49 ± 163.98	332.55 ± 0.13	584.06 ± 196.36	282.79 ± 0.13	**286.76 ± 263.86**	474.07 ± 0.13
state_waveletLast	1949.75 ± 246.48	558.22 ± 0.14	1098.09 ± 153.14	514.98 ± 0.15	519.78 ± 169.83	1037.17 ± 0.19
prefix_waveletAgg	9303.17 ± 260.73	217.71 ± 219.76	19973.74 ± 258.86	121.63 ± 202.12	2697.47 ± 227.16	**127.37 ± 215.01**
cluster_WaveletAgg	490300.2 ± 254.97	606.78 ± 0.59	347419.64 ± 185.69	543.62 ± 0.63	16230344.07 ± 237.47	444.66 ± 0.35
prefix_waveletIndex	650.21 ± 161.9	273.26 ± 256.63	723.74 ± 161.31	237.54 ± 180.03	588.79 ± 227.72	196.7 ± 151.53

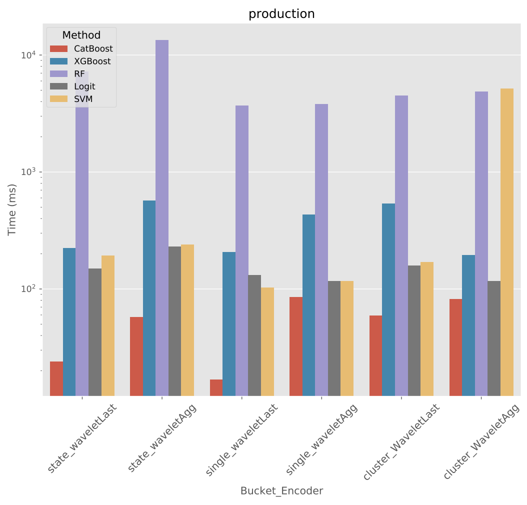
	bpic2015_1		**bpic2015_5**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	523.61 ± 270.34	280.27 ± 0.14	939.8 ± 122.33	625.94 ± 0.34	11604.54 ± 213.63	367.33 ± 0.14
prefix_waveletLast	7700.86 ± 251.72	258.81 ± 117.73	11469.51 ± 194.73	185.91 ± 277.39	1535.62 ± 125.99	272.87 ± 197.95
state_waveletAgg	**368.28 ± 203.37**	111.72 ± 0.03	835.62 ± 243.37	136.71 ± 0.04	26407.19 ± 153.89	377.12 ± 0.09
single_waveletLast	458.79 ± 258.26	373.7 ± 0.11	3461.6 ± 117.93	334.89 ± 0.1	715.89 ± 199.41	326.33 ± 0.07
single_waveletAgg	1050.34 ± 200.21	439.32 ± 0.16	696.23 ± 171.81	1203.48 ± 0.47	7121.49 ± 145.84	412.8 ± 0.14
state_waveletLast	682.51 ± 281.94	**93.05 ± 0.02**	**567.06 ± 251.81**	**83.54 ± 0.02**	**186.68 ± 218.1**	814.01 ± 0.17
prefix_waveletAgg	43496.73 ± 176.34	195.83 ± 239.63	66276.38 ± 196.92	199.86 ± 202.66	7953.67 ± 248.62	**179.54 ± 156.01**
cluster_WaveletAgg	14822144.6 ± 260.49	171.33 ± 0.18	107739.09 ± 224.38	189.09 ± 0.2	92412515.94 ± 259.41	325.06 ± 0.29
prefix_waveletIndex	1453.76 ± 123.45	277.2 ± 194.6	2958.93 ± 216.19	248.74 ± 184.14	598.28 ± 179.55	220.35 ± 165.28

	bpic2017_C		**production**		**sepsis_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	61082.52 ± 188.56	390.03 ± 0.15	789.61 ± 252.03	363.3 ± 0.12	139.11 ± 221.77	346.75 ± 0.13
prefix_waveletLast	15968.85 ± 179.11	182.32 ± 221.64	587.73 ± 259.78	160.89 ± 268.31	1538.88 ± 255.52	226.98 ± 271.71
state_waveletAgg	4288.49 ± 194.92	338.23 ± 0.09	624.55 ± 123.03	576.37 ± 0.1	**59.55 ± 124.3**	266.36 ± 0.06
single_waveletLast	3071.76 ± 142.65	165.01 ± 0.04	548.47 ± 206.46	186.46 ± 0.03	1816.56 ± 186.36	352.98 ± 0.09
single_waveletAgg	12875.43 ± 202.23	230.2 ± 0.07	623.95 ± 220.9	723.66 ± 0.19	547.09 ± 250.6	414.94 ± 0.13
state_waveletLast	**1397.02 ± 187.12**	420.04 ± 0.09	**75.97 ± 252.05**	275.49 ± 0.05	1994.32 ± 131.31	720.08 ± 0.14
prefix_waveletAgg	117523.97 ± 208.69	232.27 ± 262.43	3008.13 ± 204.27	**113.45 ± 221.71**	7646.81 ± 265.2	**149.46 ± 161.04**
cluster_WaveletAgg	648895.34 ± 162.56	207.55 ± 0.17	178121.32 ± 244.74	248.71 ± 0.17	457006.57 ± 187.91	400.56 ± 0.32
prefix_waveletIndex	23846.84 ± 153.15	**129.73 ± 282.94**	138.61 ± 150.66	181.97 ± 233.41	565.18 ± 160.58	152.26 ± 202.57

	bpic2011_1		**bpic2017_R**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	4811.74 ± 118.69	594.04 ± 0.3	4747.1 ± 119.74	296.12 ± 0.12
prefix_waveletLast	2755.1 ± 121.75	264.29 ± 171.89	23158.02 ± 230.27	239.75 ± 228.36
state_waveletAgg	9285.52 ± 239.27	422.79 ± 0.12	48231.41 ± 161.74	303.94 ± 0.08
single_waveletLast	390.57 ± 185.01	**103.33 ± 0.03**	28447.68 ± 224.19	160.66 ± 0.04
single_waveletAgg	**198.31 ± 238.35**	324.07 ± 0.15	48869.95 ± 252.21	228.46 ± 0.06
state_waveletLast	2662.42 ± 253.82	567.85 ± 0.15	**1953.79 ± 159.94**	225.39 ± 0.05
prefix_waveletAgg	12826.59 ± 243.19	144.5 ± 208.46	135909.32 ± 146.77	**128.59 ± 121.59**
cluster_WaveletAgg	372658.92 ± 269.57	373.94 ± 0.28	42163742.58 ± 138.34	153.3 ± 0.13
prefix_waveletIndex	655.82 ± 228.91	283.05 ± 137.36	26107.58 ± 140.64	159.81 ± 246.33



(a) Without wavelet



(b) With wavelet

Figure 5.29: Execution time before and after adding the wavelet encoding method (online) on Production event log

5.4.3 Inter-case features results

Adding inter-case features to the original event logs has shown a significant improvement in terms of accuracy. To measure time performance after adding inter-case features, we excluded the time taken to extract these features from the event logs; however, for massive logs, the feature extraction process took much time. Due to the time and resource constraints, we ran all experiments only one time using the best hyperparameter from previous tests, and we report the execution time for each sequence encoding, trace bucketing, and classification method separately.

Table 5.10 shows the execution time of inter-case features with CatBoost classifier. Tables B.21 (XGBoost), B.22 (Logit), B.23 (RF), and B.24 (SVM).

5.5 Summary

In this Chapter, we evaluated our proposed methods w.r.t other existing predictive monitoring approaches. The experiments have performed across 20 different predictive tasks from 7 real-life data sets. The evaluation shows that our three proposed methods have improved the overall prediction accuracy on average, however, w.r.t., the execution time all results show that two out of the three proposed methods do training

Table 5.10: Execution times for **inter-case features plus CatBoost**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	184.26 ± 0.0	0.02 ± 0.03	409.6 ± 0.0	0.01 ± 0.01	40.38 ± 0.0	0.01 ± 0.02
cluster_agg	130.73 ± 0.0	0.01 ± 0.02	392.35 ± 0.0	0.01 ± 0.01	-	-
single_agg	84.29 ± 0.0	0.01 ± 0.01	374.17 ± 0.0	0.01 ± 0.01	113.69 ± 0.0	0.01 ± 0.01
prefix_agg	0.83 ± 0.0	**0.0 ± 0.0**	2.5 ± 0.0	**0.0 ± 0.0**	0.61 ± 0.0	**0.0 ± 0.0**
prefix_laststate	0.85 ± 0.0	**0.0 ± 0.0**	2.57 ± 0.0	**0.0 ± 0.0**	0.62 ± 0.0	**0.0 ± 0.0**
single_laststate	47.06 ± 0.0	**0.0 ± 0.01**	252.86 ± 0.0	**0.0 ± 0.01**	146.15 ± 0.0	**0.0 ± 0.01**
prefix_index	**0.13 ± 0.0**	**0.0 ± 0.0**	**0.73 ± 0.0**	**0.0 ± 0.0**	**0.1 ± 0.0**	**0.0 ± 0.0**
state_agg	626.04 ± 0.0	0.01 ± 0.01	4288.44 ± 0.0	0.01 ± 0.01	756.47 ± 0.0	0.01 ± 0.01
state_laststate	270.68 ± 0.0	**0.0 ± 0.01**	190.0 ± 0.0	**0.0 ± 0.01**	168.71 ± 0.0	**0.0 ± 0.01**

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	258.78 ± 0.0	0.01 ± 0.01	1432.13 ± 0.0	0.02 ± 0.02	102.57 ± 0.0	0.03 ± 0.04
cluster_agg	306.55 ± 0.0	0.01 ± 0.01	889.51 ± 0.0	0.02 ± 0.02	491.4 ± 0.0	0.03 ± 0.05
single_agg	226.03 ± 0.0	0.01 ± 0.01	2596.3 ± 0.0	0.01 ± 0.01	535.87 ± 0.0	0.01 ± 0.01
prefix_agg	2.47 ± 0.0	**0.0 ± 0.0**	7.39 ± 0.0	**0.0 ± 0.0**	1.17 ± 0.0	**0.0 ± 0.0**
prefix_laststate	2.49 ± 0.0	**0.0 ± 0.0**	6.74 ± 0.0	**0.0 ± 0.0**	1.16 ± 0.0	**0.0 ± 0.0**
single_laststate	-	-	103.5 ± 0.0	0.01 ± 0.01	111.64 ± 0.0	**0.0 ± 0.01**
prefix_index	**0.73 ± 0.0**	**0.0 ± 0.0**	**5.32 ± 0.0**	**0.0 ± 0.0**	**0.19 ± 0.0**	**0.0 ± 0.0**
state_agg	717.6 ± 0.0	0.01 ± 0.01	2350.33 ± 0.0	0.01 ± 0.01	475.98 ± 0.0	0.01 ± 0.01
state_laststate	-	-	856.39 ± 0.0	0.01 ± 0.01	1807.14 ± 0.0	0.01 ± 0.01

	bpic2012_A		**bpic2015_3**		**bpic2011_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	327.9 ± 0.0	0.01 ± 0.01	138.26 ± 0.0	0.01 ± 0.02	48.97 ± 0.0	0.03 ± 0.05
cluster_agg	267.41 ± 0.0	0.01 ± 0.01	1183.33 ± 0.0	0.01 ± 0.02	204.86 ± 0.0	0.03 ± 0.05
single_agg	326.55 ± 0.0	0.01 ± 0.01	196.07 ± 0.0	0.01 ± 0.01	429.3 ± 0.0	0.01 ± 0.01
prefix_agg	2.57 ± 0.0	**0.0 ± 0.0**	1.33 ± 0.0	**0.0 ± 0.0**	0.6 ± 0.0	**0.0 ± 0.0**
prefix_laststate	2.48 ± 0.0	**0.0 ± 0.0**	1.35 ± 0.0	**0.0 ± 0.0**	0.61 ± 0.0	**0.0 ± 0.0**
single_laststate	251.24 ± 0.0	**0.0 ± 0.01**	177.26 ± 0.0	**0.0 ± 0.01**	119.47 ± 0.0	0.01 ± 0.01
prefix_index	**0.71 ± 0.0**	**0.0 ± 0.0**	**0.22 ± 0.0**	**0.0 ± 0.0**	**0.18 ± 0.0**	**0.0 ± 0.0**
state_agg	1422.77 ± 0.0	0.01 ± 0.01	2938.17 ± 0.0	0.01 ± 0.01	1451.24 ± 0.0	0.01 ± 0.01
state_laststate	297.06 ± 0.0	**0.0 ± 0.01**	5487.9 ± 0.0	**0.0 ± 0.01**	146.52 ± 0.0	0.01 ± 0.01

	sepsis_2		**bpic2011_2**		**bpic2017_C**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	143.47 ± 0.0	0.03 ± 0.03	83.32 ± 0.0	0.03 ± 0.05	618.74 ± 0.0	0.02 ± 0.03
cluster_agg	190.76 ± 0.0	0.03 ± 0.03	141.94 ± 0.0	0.03 ± 0.04	1316.23 ± 0.0	0.02 ± 0.02
single_agg	46.92 ± 0.0	0.01 ± 0.01	97.1 ± 0.0	0.01 ± 0.01	1173.71 ± 0.0	0.01 ± 0.01
prefix_agg	0.16 ± 0.0	**0.0 ± 0.0**	1.26 ± 0.0	**0.0 ± 0.0**	6.93 ± 0.0	**0.0 ± 0.0**
prefix_laststate	0.16 ± 0.0	**0.0 ± 0.0**	1.26 ± 0.0	**0.0 ± 0.0**	8.53 ± 0.0	**0.0 ± 0.0**
single_laststate	15.12 ± 0.0	0.01 ± 0.01	148.72 ± 0.0	0.01 ± 0.01	692.22 ± 0.0	0.01 ± 0.01
prefix_index	**0.12 ± 0.0**	**0.0 ± 0.0**	**0.2 ± 0.0**	**0.0 ± 0.0**	**6.07 ± 0.0**	**0.0 ± 0.0**
state_agg	324.12 ± 0.0	0.02 ± 0.02	1601.03 ± 0.0	0.01 ± 0.01	1970.88 ± 0.0	0.01 ± 0.01
state_laststate	114.61 ± 0.0	0.01 ± 0.03	2431.37 ± 0.0	0.01 ± 0.01	900.71 ± 0.0	0.01 ± 0.01

	bpic2015_5		**sepsis_1**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	87.35 ± 0.0	0.02 ± 0.02	1228.41 ± 0.0	0.02 ± 0.03	261.55 ± 0.0	0.02 ± 0.03
cluster_agg	-	-	78.15 ± 0.0	0.03 ± 0.03	30.57 ± 0.0	0.03 ± 0.03
single_agg	146.47 ± 0.0	0.01 ± 0.01	105.84 ± 0.0	0.01 ± 0.01	99.36 ± 0.0	0.01 ± 0.01
prefix_agg	1.09 ± 0.0	**0.0 ± 0.0**	0.49 ± 0.0	**0.0 ± 0.0**	0.52 ± 0.0	**0.0 ± 0.0**
prefix_laststate	1.26 ± 0.0	**0.0 ± 0.0**	0.48 ± 0.0	**0.0 ± 0.0**	0.52 ± 0.0	**0.0 ± 0.0**
single_laststate	26.64 ± 0.0	**0.0 ± 0.01**	15.73 ± 0.0	0.01 ± 0.01	72.05 ± 0.0	0.01 ± 0.01
prefix_index	**0.17 ± 0.0**	**0.0 ± 0.0**	**0.12 ± 0.0**	**0.0 ± 0.0**	**0.12 ± 0.0**	**0.0 ± 0.0**
state_agg	892.39 ± 0.0	0.01 ± 0.01	442.88 ± 0.0	0.01 ± 0.01	96.79 ± 0.0	0.01 ± 0.02
state_laststate	3813.72 ± 0.0	**0.0 ± 0.01**	783.33 ± 0.0	0.01 ± 0.01	75.67 ± 0.0	0.01 ± 0.02

	bpic2015_1		**production**		**bpic2011_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	39.51 ± 0.0	0.02 ± 0.03	53.36 ± 0.0	0.03 ± 0.03	408.88 ± 0.0	0.03 ± 0.04
cluster_agg	80.61 ± 0.0	0.02 ± 0.02	28.11 ± 0.0	0.02 ± 0.02	659.43 ± 0.0	0.03 ± 0.05
single_agg	578.36 ± 0.0	0.01 ± 0.01	25.98 ± 0.0	0.02 ± 0.01	82.14 ± 0.0	0.01 ± 0.01
prefix_agg	0.74 ± 0.0	**0.0 ± 0.0**	0.12 ± 0.0	**0.0 ± 0.0**	0.82 ± 0.0	**0.0 ± 0.0**
prefix_laststate	0.74 ± 0.0	**0.0 ± 0.0**	0.12 ± 0.0	**0.0 ± 0.0**	0.83 ± 0.0	**0.0 ± 0.0**
single_laststate	347.68 ± 0.0	**0.0 ± 0.01**	5.56 ± 0.0	0.01 ± 0.01	32.88 ± 0.0	0.01 ± 0.01
prefix_index	**0.11 ± 0.0**	**0.0 ± 0.0**	**0.04 ± 0.0**	**0.0 ± 0.0**	**0.19 ± 0.0**	**0.0 ± 0.0**
state_agg	338.81 ± 0.0	0.01 ± 0.01	56.01 ± 0.0	0.02 ± 0.02	2844.76 ± 0.0	0.01 ± 0.01
state_laststate	523.99 ± 0.0	**0.0 ± 0.01**	22.45 ± 0.0	0.01 ± 0.01	1463.34 ± 0.0	0.01 ± 0.01

	bpic2017_R	
method	training_total (s)	prediction_avg (ms)
cluster_laststate	993.7 ± 0.0	0.02 ± 0.02
cluster_agg	390.33 ± 0.0	0.02 ± 0.02
single_agg	564.28 ± 0.0	0.01 ± 0.01
prefix_agg	7.59 ± 0.0	**0.0 ± 0.0**
prefix_laststate	6.89 ± 0.0	**0.0 ± 0.0**
single_laststate	893.97 ± 0.0	0.02 ± 0.03
prefix_index	**5.45 ± 0.0**	**0.0 ± 0.0**
state_agg	1296.68 ± 0.0	0.01 ± 0.01
state_laststate	842.36 ± 0.0	0.01 ± 0.01

and prediction in a reasonable time, and only the wavelet encoding method takes an enormous quantity time that could be improved later.

Table 5.11 contains a summary of all results that we got after adding our proposed methods to existing outcome-oriented PPM techniques as a way to improve its performance.

Table 5.11: Summary of all results that we got from our three proposed approaches to improve existing outcome-oriented PPM methods

Method	Contribution 1 (CatBoost)	Contribution 2 (Wavelet encoding)	Contribution 3 (Inter-case features)
CatBoost	**10**	1	8
XGBoost	3	5	11
Random forest (RF)	2	4	**13**
Logistic regression (LR)	1	**13**	5
Support vector machine (SVM)	3	6	10

In the beginning, we added a *CatBoost* algorithm as a new classification method to the area of outcome-oriented PPM. Results show that we have a significant improvement in 10 event logs; however, after adding wavelet encoding with CatBoost, we achieved only one improvement in 1 event log. Finally, using CatBoost with inter-case features, we got an improvement over previous methods in 7 event logs.

All classification methods show significant improvement after adding inter-case features as shown in Table 5.11, such as *XGBoost* which achieves an increase in AUC on 11 event logs over baseline or after adding wavelet encoding. Moreover, using wavelet encoding with XGBoost show a significant improvement in 5 event logs.

Random forest and *logistic regression* show an increase in performance after adding inter-case features or wavelet encoding methods in 13 event logs, respectively. Furthermore, We can achieve better performance using *support vector machine* after adding the wavelet encoding method (in 6 event logs) or inter-caste features (in 10 event logs).

6

Conclusions and Future work

6.1 Conclusion

In this thesis, we introduced three different methods to improve the outcome-oriented predictive process monitoring performance. (i) A new gradient boosting algorithm that handles categorical features and uses order boosting, i.e. *CatBoost* to the area of outcome-oriented predictive process monitoring; (ii) We proposed a new *complex encoding method* using discrete wavelet transformation and time series encoding in addition to autoencoder neural network; (iii) A new set of *inter-case features* that come from the concurrent execution of different business cases and capture information about workload, demand intensity, and the temporal context of business processes.

We evaluated our proposed methods to a baseline from the current existing predictive monitoring methods w.r.t three different evaluation metrics to quantify the goodness and the badness of our methods. Results show an improvement in outcome-oriented predictive monitoring methods in terms of prediction accuracy however one of the proposed methods (i.e. wavelet encoding) has a drawback w.r.t time performance as the time complexity is $O(n^3)$.

6.2 Future work

In recent years, the area of outcome-oriented predictive monitoring becomes a critical field for all organizations and companies as they need to improve their business process performance which mainly depends on improving the existing methods of PPM. Predictive monitoring methods depend on many factors that need more investigation to get an improvement in the outcome-oriented PPM area. (i) One of the most important factors and related to this thesis is how to encode the original event log into a feature vector using a lossless encoding way to capture all information from the log. For example, how to optimize the wavelet encoding method to be executed in a reasonable amount of time. (ii) Another factor is related to the event log itself since we considered only in this thesis a structured data, so we need to explore how the proposed methods will perform on unstructured data such as text. (iii) Diving into outcome-oriented problems to build prescriptive models that can help the business system worker with some guidance and decisions to avoid unexpected outcomes during the running of business cases. (iv) One limitation on this thesis that can be a good direction for future empirical research is we compared a combination between first contribution (CatBoost) and third contribution (inter-case features) to the baseline, but we didn't combine the second contribution (wavelet) with the third one due to the limitation of time. Accordingly, one question to answer in the future, what is the performance of wavelet encodings in combination with inter-case features?

Bibliography

[1] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, et al., Fundamentals of business process management, Vol. 1, Springer, 2018. vii, 3

[2] M. Dumas, M. Reichert, M.-C. Shan, Business process management, Springer, 2008. 2

[3] M. Weske, Business process management architectures, Business Process Management: Concepts, Languages, Architectures (2007) 305–343. 2

[4] W. Van Der Aalst, Data science in action, in: Process mining, Springer, 2016, pp. 3–23. 2

[5] I. Teinemaa, M. Dumas, M. L. Rosa, F. M. Maggi, Outcome-oriented predictive process monitoring: Review and benchmark, ACM Transactions on Knowledge Discovery from Data (TKDD) 13 (2) (2019) 1–57. 4, 7, 8, 12, 13, 16, 21, 23, 26, 28, 30, 42, 43, 48, 52

[6] A. Senderovich, C. Di Francescomarino, F. M. Maggi, From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring, Information Systems 84 (2019) 255–264. 6, 9, 26, 40, 48

[7] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, A. Gulin, Catboost: unbiased boosting with categorical features, in: Advances in neural information processing systems, 2018, pp. 6638–6648. 8, 32, 33

[8] T. Edwards, Discrete wavelet transforms: Theory and implementation, Universidad de (1991) 28–35. 8

[9] J. Bergstra, D. Yamins, D. D. Cox, Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms, in: Proceedings of the 12th Python in science conference, Citeseer, 2013, pp. 13–20. 12

[10] A. P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern recognition 30 (7) (1997) 1145–1159. 16

[11] J. Platt, et al., Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, Advances in large margin classifiers 10 (3) (1999) 61–74. 19

[12] Z. Xing, J. Pei, S. Y. Philip, Early classification on time series, Knowledge and information systems 31 (1) (2012) 105–127. 19

[13] U. Mori, A. Mendiburu, E. Keogh, J. A. Lozano, Reliable early classification of time series based on discriminating the classes over time, Data mining and knowledge discovery 31 (1) (2017) 233–263. 19

[14] T. Santos, R. Kern, A literature survey of early time series classification and deep learning., in: Sami@ iknow, 2016. 19

[15] Z. Xing, J. Pei, E. Keogh, A brief survey on sequence classification, ACM Sigkdd Explorations Newsletter 12 (1) (2010) 40–48. 19

[16] A. Leontjeva, R. Conforti, C. Di Francescomarino, M. Dumas, F. M. Maggi, Complex symbolic sequence encodings for predictive monitoring of business processes, in: International Conference on Business Process Management, Springer, 2016, pp. 297–313. 19, 20, 24, 25, 26, 28, 29, 48

[17] C. Di Francescomarino, C. Ghidini, F. M. Maggi, G. Petrucci, A. Yeshchenko, An eye into the future: leveraging a-priori knowledge in predictive business process monitoring, in: International Conference on Business Process Management, Springer, 2017, pp. 252–268. 20

[18] Y.-F. Lin, H.-H. Chen, V. S. Tseng, J. Pei, Reliable early classification on multivariate time series with numerical and categorical attributes, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2015, pp. 199–211. 20

[19] I. Verenich, M. Dumas, M. La Rosa, F. M. Maggi, C. Di Francescomarino, Complex symbolic sequence clustering and multiple classifiers for predictive process monitoring, in: International Conference on Business Process Management, Springer, 2016, pp. 218–229. 24, 29

[20] M. De Leoni, W. M. van der Aalst, M. Dees, A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs, Information Systems 56 (2016) 235–257. 24, 27, 28

[21] F. M. Maggi, C. Di Francescomarino, M. Dumas, C. Ghidini, Predictive monitoring of business processes, in: International conference on advanced information systems engineering, Springer, 2014, pp. 457–472. 25, 26, 28

[22] C. Di Francescomarino, M. Dumas, F. M. Maggi, I. Teinemaa, Clustering-based predictive process monitoring, IEEE transactions on services computing 12 (6) (2017) 896–909. 25, 30, 45

[23] D. Grigori, F. Casati, U. Dayal, M.-C. Shan, Improving business process quality through exception understanding, prediction, and prevention, in: VLDB, Vol. 1, 2001, pp. 159–168. 25, 28

[24] R. Conforti, M. de Leoni, M. La Rosa, W. M. van der Aalst, A. H. ter Hofstede, A recommendation system for predicting risks across multiple business process instances, Decision Support Systems 69 (2015) 1–19. 25, 26

[25] B. Schwegmann, M. Matzner, C. Janiesch, A method and tool for predictive event-driven process analytics., in: Wirtschaftsinformatik, 2013, p. 46. 25, 28

[26] G. T. Lakshmanan, S. Duan, P. T. Keyser, F. Curbera, R. Khalaf, Predictive analytics for semi-structured case oriented business processes, in: International Conference on Business Process Management, Springer, 2010, pp. 640–651. 25

[27] S. Van Der Spoel, M. Van Keulen, C. Amrit, Process prediction in noisy data sets: a case study in a dutch hospital, in: International Symposium on Data-Driven Process Discovery and Analysis, Springer, 2012, pp. 60–83. 25, 27, 28, 29

[28] W. M. Van der Aalst, V. Rubin, H. Verbeek, B. F. van Dongen, E. Kindler, C. W. Günther, Process mining: a two-step approach to balance between underfitting and overfitting, Software & Systems Modeling 9 (1) (2010) 87. 26

[29] J. Ghattas, P. Soffer, M. Peleg, Improving business process decision making based on past experience, Decision Support Systems 59 (2014) 93–107. 28

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, in: Advances in neural information processing systems, 2017, pp. 3146–3154. 32

[31] M. Weeks, M. Bayoumi, Discrete wavelet transform: architectures, design and performance issues, Journal of VLSI signal processing systems for signal, image and video technology 35 (2) (2003) 155–178. 35, 37

[32] F. Taymouri, M. La Rosa, J. Carmona, Business process variant analysis based on mutual fingerprints of event logs, arXiv preprint arXiv:1912.10598. 35, 36

[33] G. Strang, G. Strang, G. Strang, G. Strang, Introduction to linear algebra, Vol. 3, Wellesley-Cambridge Press Wellesley, MA, 1993. 36

[34] S. Santoso, E. J. Powers, W. M. Grady, Power quality disturbance data compression using wavelet transform methods, IEEE Transactions on Power Delivery 12 (3) (1997) 1250–1257. 36

[35] K. Rao, N. Ahmed, Orthogonal transforms for digital signal processing, in: ICASSP'76. IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 1, IEEE, 1976, pp. 136–140. 36

[36] R. H. Shumway, D. S. Stoffer, Time series analysis and its applications: with R examples, Springer, 2017. 36

[37] C. C. Aggarwal, Data mining: the textbook, Springer, 2015. 37

[38] B. R. Bakshi, Multiscale analysis and modeling using wavelets, Journal of chemometrics 13 (3-4) (1999) 415–434. 37

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830. 42

[40] 4tu centre for research data, https://researchdata.4tu.nl/. 43

[41] A. Pnueli, The temporal logic of programs, in: 18th Annual Symposium on Foundations of Computer Science (sfcs 1977), IEEE, 1977, pp. 46–57. 45

[42] C. W. Günther, E. Verbeek, Xes standard definition, Fluxicon Process Laboratories (November 2009). 45

[43] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, Data preprocessing for supervised leaning, International Journal of Computer Science 1 (2) (2006) 111–117. 45

[44] T. M. Mitchell, et al., Machine learning (1997). 46

[45] A. Rozumnyi, A dashboard-based predictive process monitoring engine, Ph.D. thesis, Masters thesis. University of Tartu (2017). 48

[46] P. Cudre-Mauroux, P. Ceravolo, D. Gaševic, Data-driven process discovery and analysis. 48

[47] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine learning research 7 (Jan) (2006) 1–30. 52, 65

[48] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in neural information processing systems, 2017, pp. 4765–4774. 70

[49] S. M. Lundberg, G. G. Erion, S.-I. Lee, Consistent individualized feature attribution for tree ensembles, arXiv preprint arXiv:1802.03888. 70

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Mahmoud Kamel Shoush**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Improved Classifier Training Methods for Predictive Process Monitoring,
   supervised by Marlon Dumas.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Mahmoud Kamel Shoush
15/05/2020

Appendix A

Appendix A. Source Code Repository

We uploaded the implementation of our proposed methods in a Github repository to reproduce the results and to make it public for further improvements.

Github link: Predictive process monitoring code

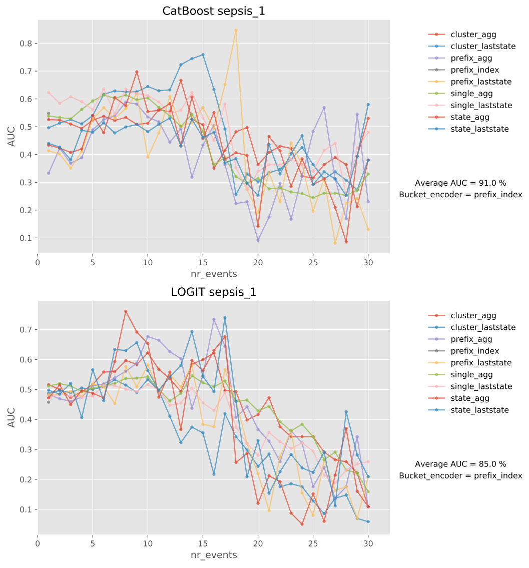
Appendix B

Appendix B. Results

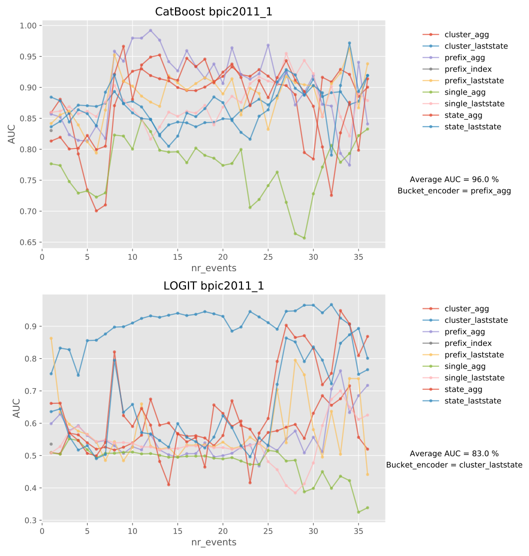
This Appendix contains all results and figures that are related to our experiments, and it organized as follows:

- All results w.r.t our first contribution ***(CatBoost)***.
  - Comparison between CatBoost and all other classification methods (i.e. Logit(B.1), RF (B.4), and SVM (B.7)).
  - Overall AUC and F-score w.r.t XGBoost (B.1), Logit (B.2), RF (B.3), and SVM (B.4)
  - Execution time for XGBoost (B.5), Logit (B.6), RF (B.7), and SVM (B.8).
- All results w.r.t our second contribution ***(Wavelet plus CatBoost)***.
  - Results before and after adding Wavelet encoding methods for CatBoost (B.10), Logit (B.13), RF (B.16), and SVM (B.19).
  - Overall AUC and F-score w.r.t XGBoost (B.9), RF (B.11), Logit (B.10), and SVM (B.12).
  - Execution time for XGBoost (B.13), Logit (B.15), RF (B.14), and SVM (B.16)
- All results w.r.t our third contribution ***(Inter-case features)***.
  - Results before and after adding inter-case features for CatBoost (B.22), XG-Boost (B.25), Logit (B.28), RF, and SVM (B.31)

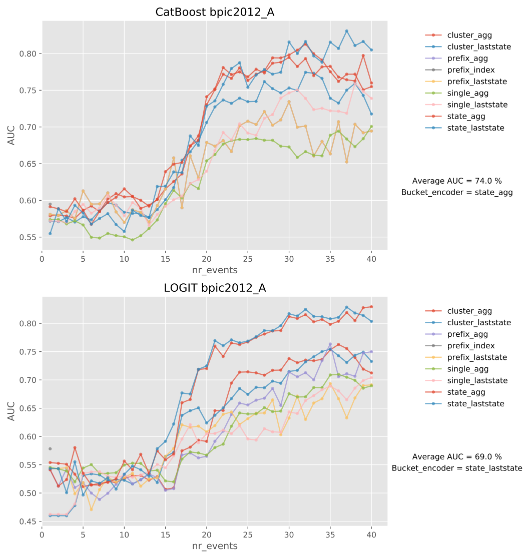
- Overall AUC and F-score w.r.t XGBoost (B.17), Logit (B.19), RF (B.18), and SVM. (B.20).
- Execution time for XGBoost (B.21), Logit (B.22), RF (B.23), and SVM (B.24).

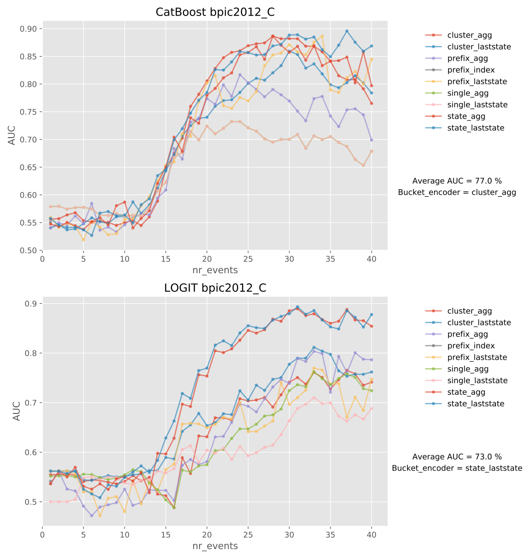


(a) CatBoost vs Logit sepsis



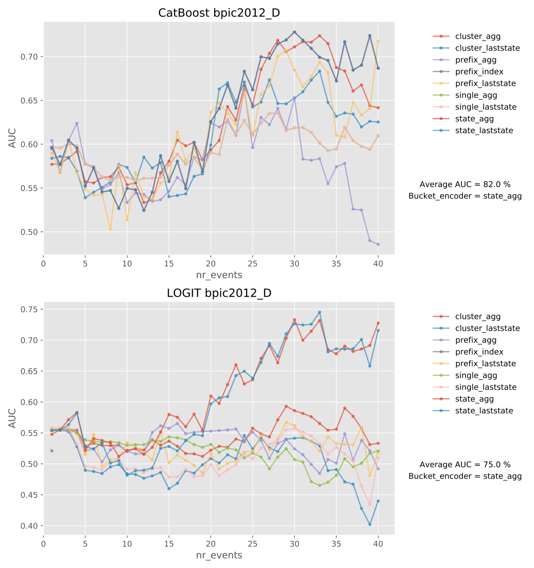
(b) CatBoost vs Logit bpic2011



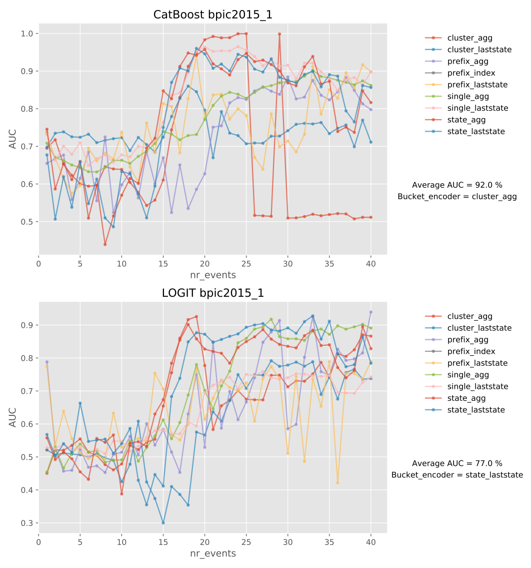


(c) CatBoost vs Logit bpic2012_A **(d)** CatBoost vs logit bpic2012_C

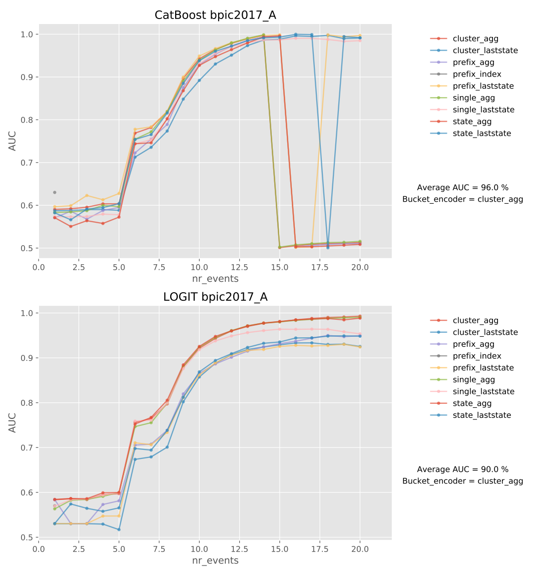
Figure B.1: Comparison between CatBoost and Logit on all event logs



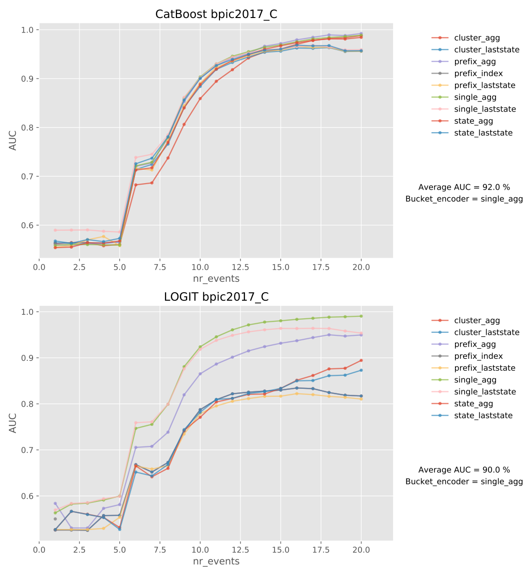
(a) CatBoost vs XGBoost bpic2012_D



(b) CatBoost vs Logit bpic2015

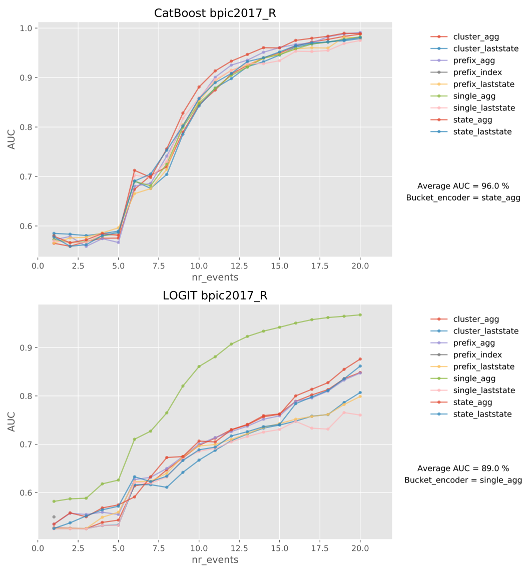


(c) CatBoost vs Logit bpic2017_A

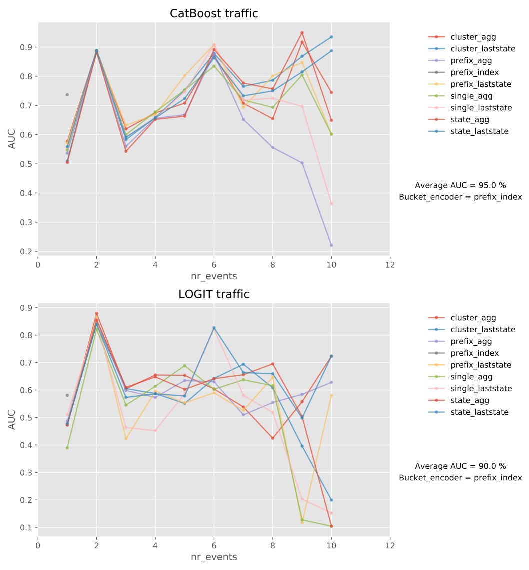


(d) CatBoost vs Logit bpic2017_C

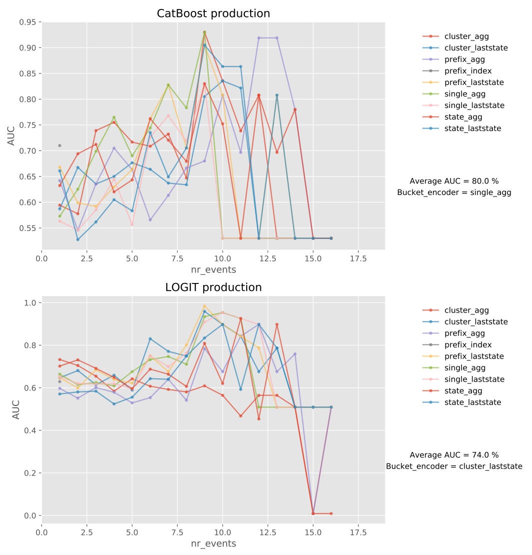
Figure B.2: Comparison between CatBoost and Logit on all event logs *(continued)*



(a) CatBoost vs Logit bpic2017_R

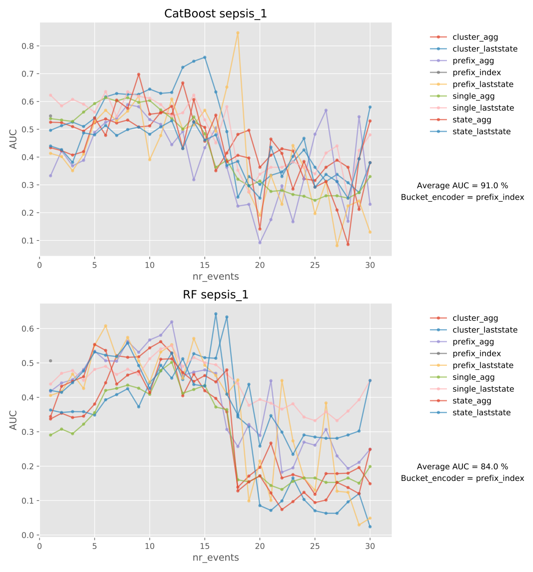


(b) CatBoost vs Logit traffic

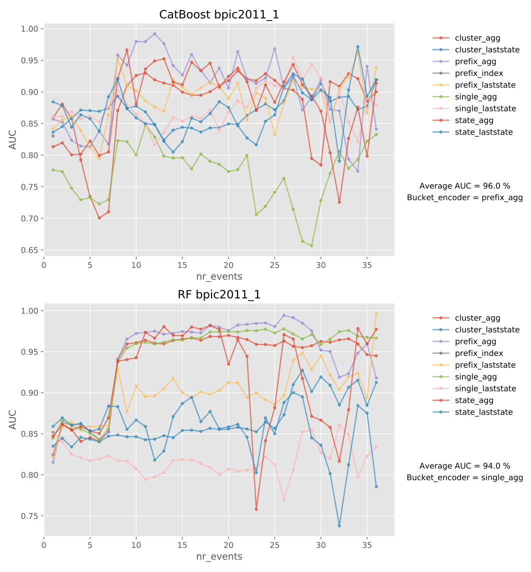


(c) CatBoost vs Logit Production

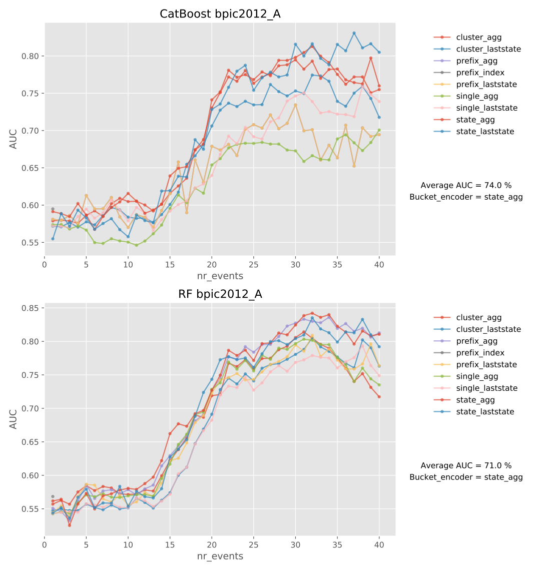
Figure B.3: Comparison between CatBoost and Logit on all event logs *(continued)*



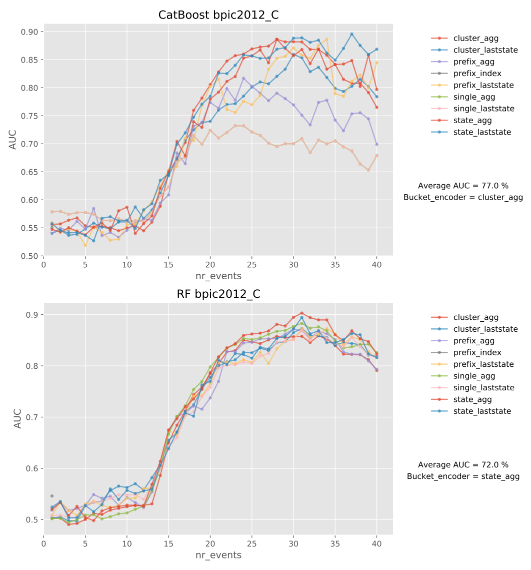
(a) CatBoost vs RF sepsis



(b) CatBoost vs RF bpic2011

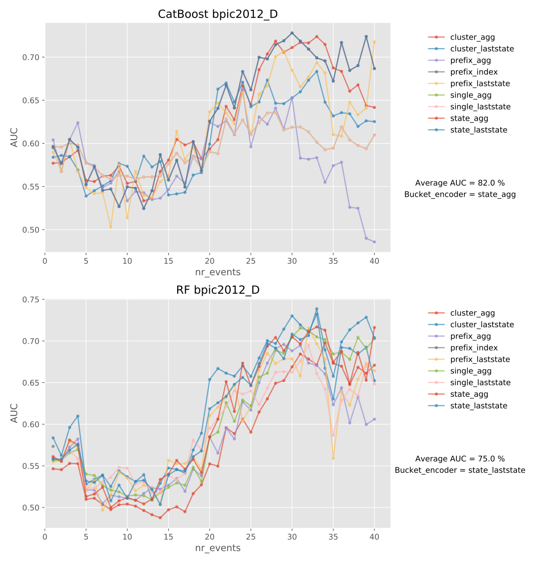


(c) CatBoost vs RF bpic2012_A

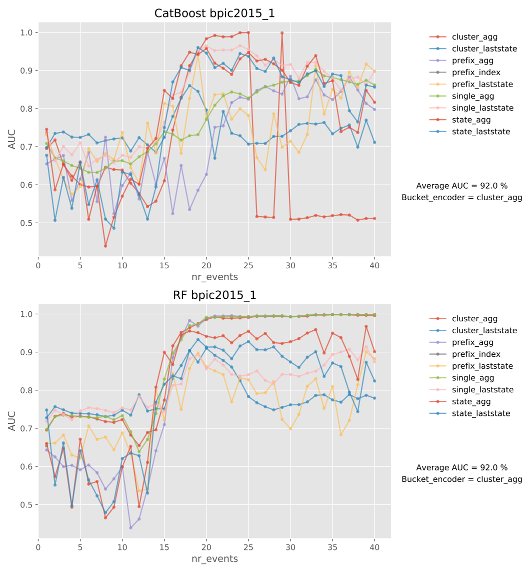


(d) CatBoost vs RF bpic2012_C

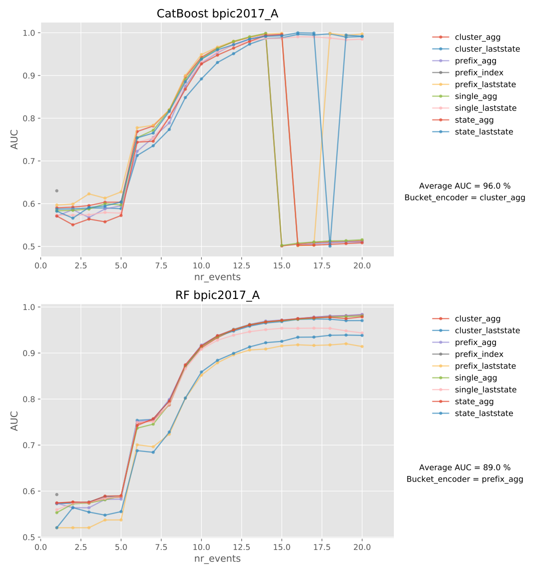
Figure B.4: Comparison between CatBoost and RF on all event logs



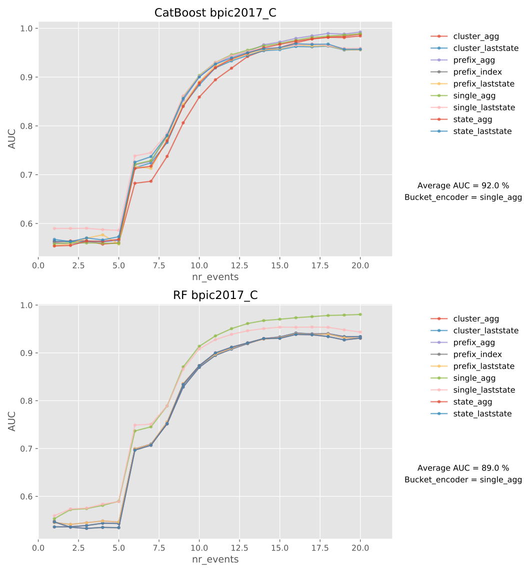
(a) CatBoost vs RF bpic2012_D



(b) CatBoost vs RF bpic2015

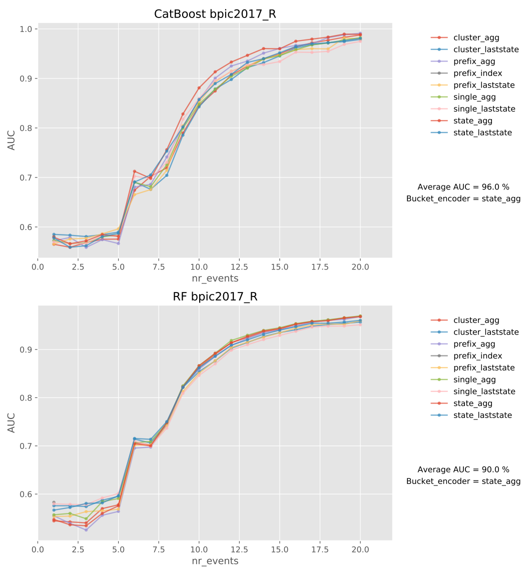


(c) CatBoost vs RF bpic2017_A

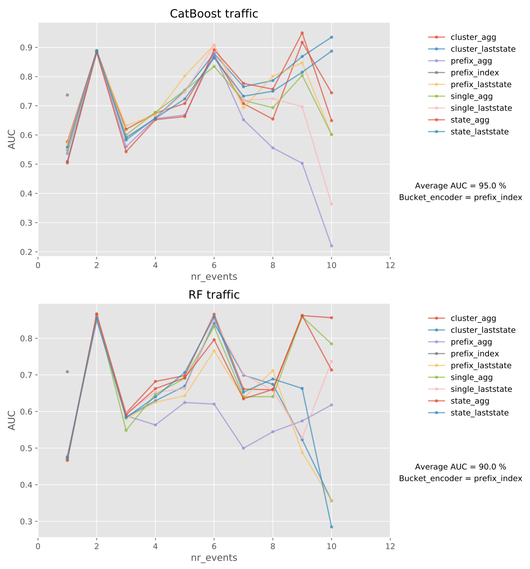


(d) CatBoost vs RF bpic2017_C

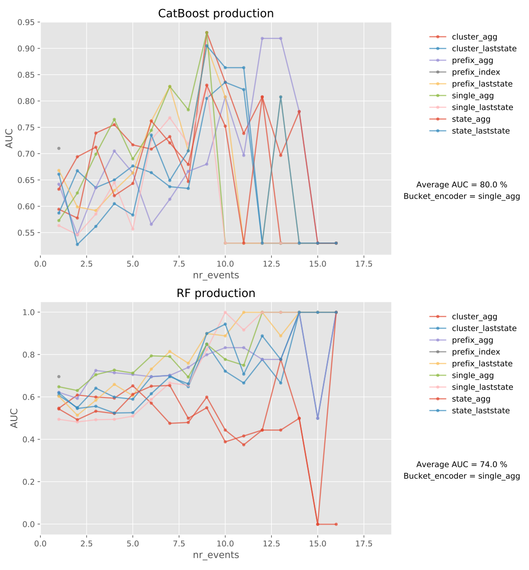
Figure B.5: Comparison between CatBoost and RF on all event logs *(continued)*.



(a) CatBoost vs RF bpic2017_R

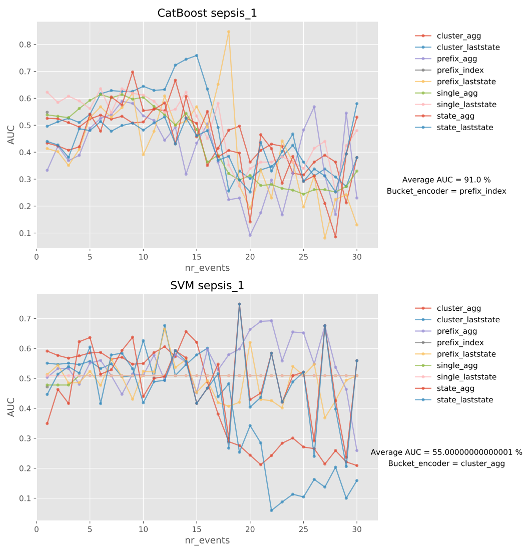


(b) CatBoost vs RF traffic

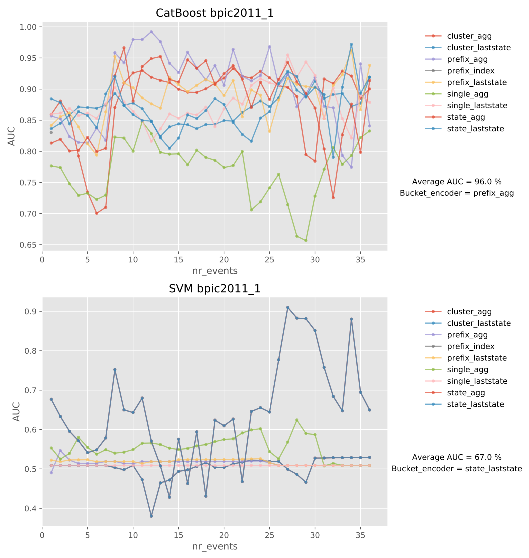


(c) CatBoost vs RF Production

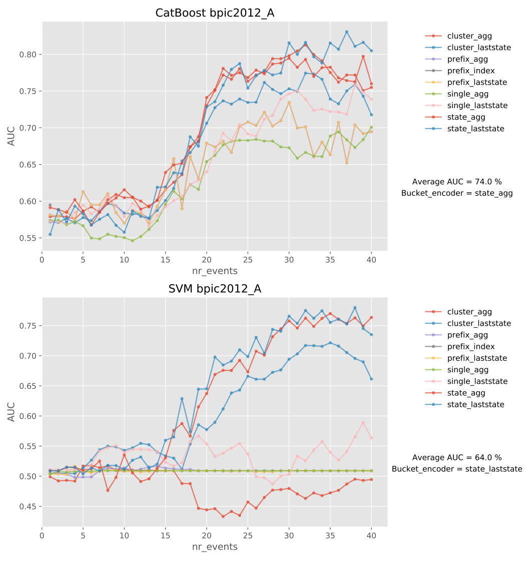
Figure B.6: Comparison between CatBoost and RF on all event logs *(continued)*.



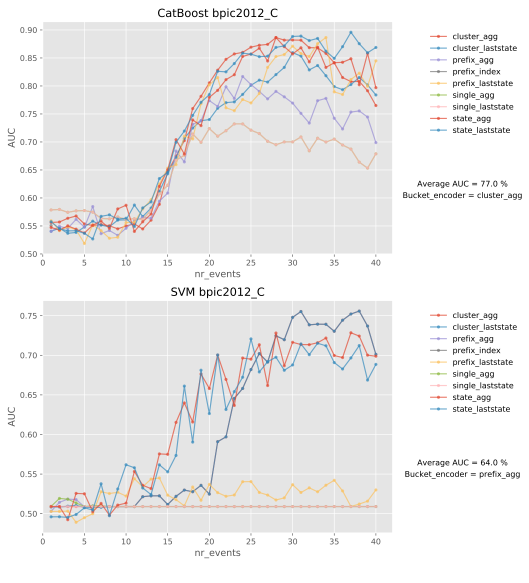
(a) CatBoost vs SVM sepsis



(b) CatBoost vs SVM bpic2011

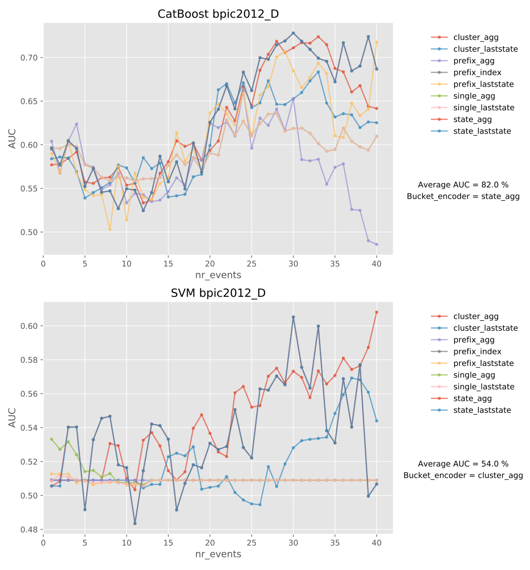


(c) CatBoost vs SVM bpic2012_A

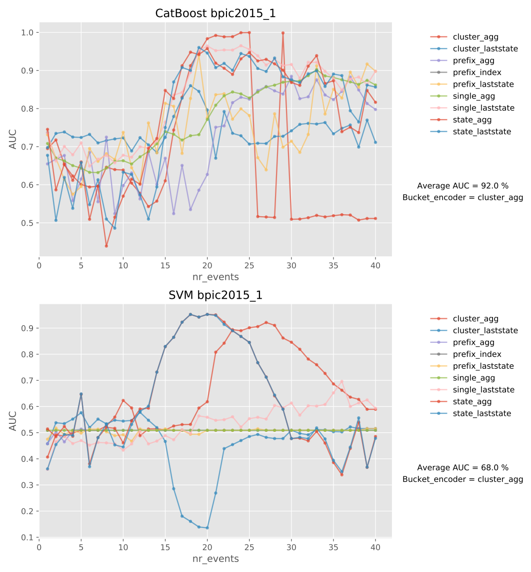


(d) CatBoost vs SVM bpic2012_C

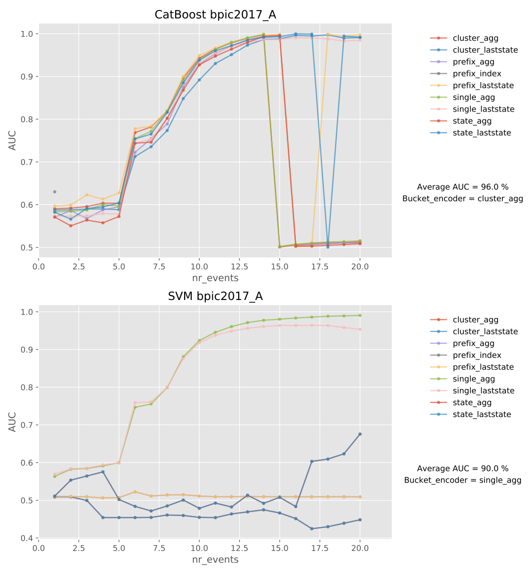
Figure B.7: Comparison between CatBoost and SVM on all event logs



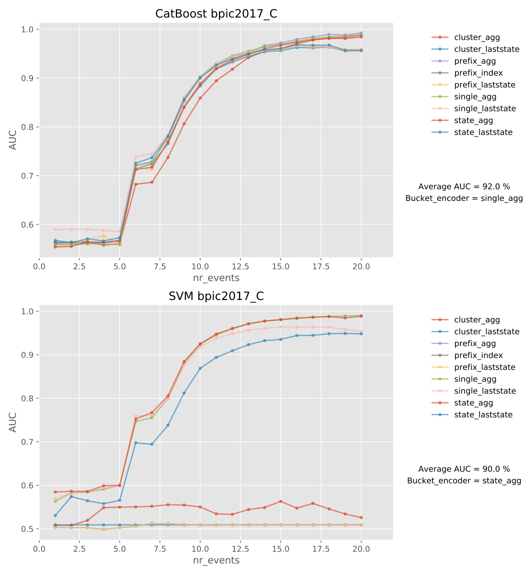
(a) CatBoost vs SVM bpic2012_D



(b) CatBoost vs SVM bpic2015

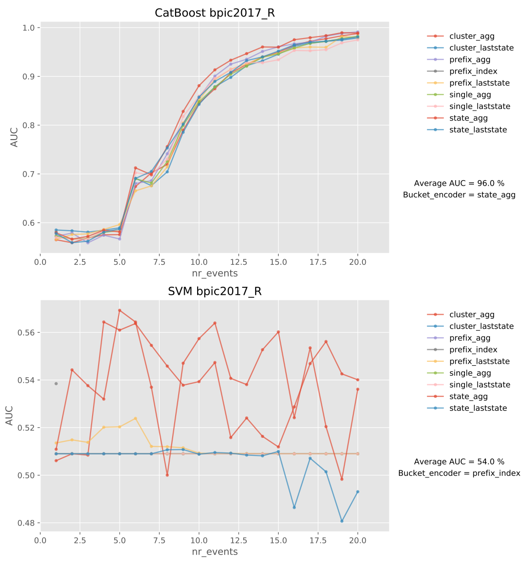


(c) CatBoost vs SVM bpic2017_A

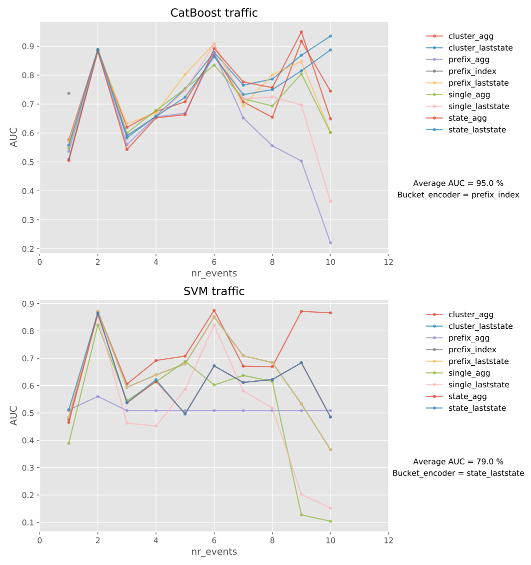


(d) CatBoost vs SVM bpic2017_C

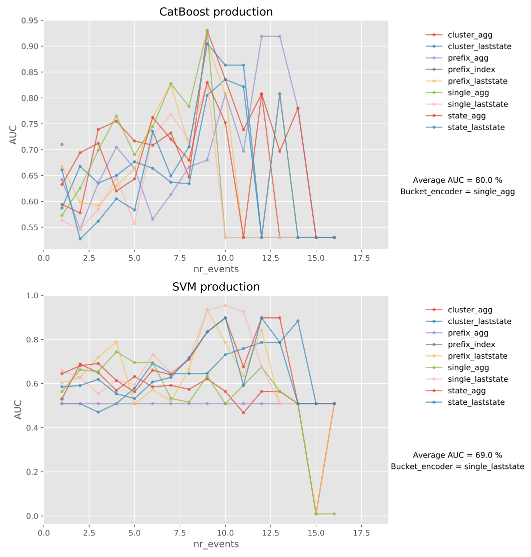
Figure B.8: Comparison between CatBoost and SVM on all event logs *(continued)*



(a) CatBoost vs SVM bpic2017_R



(b) CatBoost vs SVM traffic



(c) CatBoost vs SVM Production

Figure B.9: Comparison between CatBoost and SVM on all event logs *(continued)*

Table B.1: Overall AUC (F-score) for **XGBoost**

	xgboost																			
	bpic2012_a	bpic2011_4	bpic2012_d	bpic2012_c	production	bpic2015_4	bpic2015_2	traffic	sepsis_3	sepsis_2	bpic2011_1	bpic2015_3	bpic2015_1	bpic2011_2	bpic2011_3	bpic2017_a	bpic2015_5	bpic2017_r	bpic2017_c	sepsis_1
single_agg	0.69 (0.61)	0.85 (0.66)	0.59 (0.19)	**0.72** (**0.48**)	0.68 (0.6)	0.62 (0.2)	0.53 (0.5)	0.67 (0.69)	0.67 (0.24)	0.53 (0.03)	0.73 (0.85)	0.64 (0.31)	0.64 (0.59)	0.52 (0.96)	0.88 (0.73)	0.73 (0.75)	0.68 (0.39)	0.81 (0.42)	0.81 (0.74)	0.47 (0.12)
state_laststate	0.69 (0.65)	0.75 (0.64)	**0.62** (**0.25**)	0.7 (0.46)	0.57 (0.54)	0.69 (0.27)	0.87 (0.33)	0.67 (0.67)	0.71 (0.27)	0.36 (0.03)	0.76 (0.69)	0.6 (0.28)	0.78 (0.5)	0.54 (0.76)	0.83 (0.58)	**0.85** (**0.74**)	0.66 (0.46)	0.8 (0.47)	0.82 (0.76)	0.45 (0.11)
cluster_laststate	0.66 (0.64)	0.77 (0.59)	0.59 (0.19)	0.7 (0.45)	0.61 (0.54)	0.69 (0.23)	0.9 (0.45)	0.69 (0.69)	0.73 (0.3)	0.53 (0.03)	0.86 (0.77)	0.59 (0.21)	0.73 (0.38)	0.84 (0.83)	0.9 (0.75)	**0.85** (**0.7**)	0.68 (0.44)	0.8 (0.46)	0.81 (0.74)	0.44 (0.07)
prefix_laststate	0.65 (0.63)	0.81 (0.66)	0.58 (0.19)	0.68 (0.44)	0.65 (0.53)	0.68 (0.24)	0.83 (0.43)	0.68 (0.69)	0.7 (0.3)	0.53 (0.03)	0.85 (0.75)	0.63 (0.27)	0.73 (0.39)	0.83 (0.82)	0.87 (0.72)	0.83 (0.73)	0.68 (0.45)	0.81 (0.48)	0.81 (0.74)	0.52 (0.1)
prefix_agg	0.67 (0.66)	0.84 (0.66)	0.59 (0.2)	0.67 (0.44)	**0.73** (**0.59**)	0.66 (0.22)	0.53 (0.5)	0.66 (0.63)	0.69 (0.18)	0.53 (0.03)	0.88 (0.81)	0.64 (0.3)	0.63 (0.57)	0.99 (0.88)	0.91 (0.69)	0.73 (0.75)	0.69 (0.47)	0.8 (0.49)	0.81 (0.74)	0.46 (0.09)
state_agg	**0.7** (**0.66**)	0.77 (0.56)	**0.62** (**0.22**)	0.71 (0.48)	0.7 (0.62)	0.64 (0.19)	0.88 (0.37)	0.66 (0.66)	0.68 (0.03)	0.36 (0.03)	0.9 (0.81)	0.63 (0.29)	0.8 (0.53)	0.64 (0.76)	0.83 (0.61)	0.7 (0.76)	0.66 (0.43)	0.81 (0.49)	0.81 (0.75)	0.5 (0.13)
cluster_agg	0.68 (0.64)	0.68 (0.55)	0.6 (0.18)	0.71 (0.45)	0.7 (0.58)	0.68 (0.27)	0.81 (0.61)	0.67 (0.67)	0.65 (0.09)	0.53 (0.03)	0.92 (0.86)	0.62 (0.37)	0.67 (0.6)	0.52 (0.99)	0.89 (0.66)	0.7 (0.73)	0.65 (0.41)	**0.82** (**0.47**)	0.81 (0.74)	0.41 (0.07)
prefix_index	0.61 (0.5)	0.8 (0.61)	0.6 (0.14)	0.56 (0.35)	0.67 (0.67)	0.68 (0.26)	0.75 (0.4)	0.7 (0.2)	0.69 (0.18)	**0.89** (**0.56**)	0.75 (0.69)	0.64 (0.34)	0.55 (0.28)	0.82 (0.86)	0.85 (0.69)	0.58 (0.45)	0.64 (0.46)	0.61 (0.06)	0.55 (0.47)	**0.57** (**0.03**)
single_laststate	0.65 (0.64)	0.79 (0.63)	**0.62** (**0.18**)	0.71 (0.46)	0.61 (0.57)	0.72 (0.24)	0.85 (0.46)	0.67 (0.67)	0.71 (0.13)	0.53 (0.03)	0.84 (0.77)	0.58 (0.33)	0.71 (0.42)	0.85 (0.85)	0.86 (0.67)	**0.85** (**0.73**)	0.7 (0.45)	0.81 (0.44)	0.8 (0.74)	0.42 (0.09)

Table B.2: Overall AUC (F-score) for **Logit**

	logit					
	bpic2012_a	bpic2011_4	bpic2012_d	bpic2012_c	production	bpic2015_4
single_agg	0.59 (0.44)	0.79 (0.66)	0.53 (0.01)	0.6 (0.1)	0.67 (0.54)	0.63 (0.01)
state_laststate	0.66 (0.44)	0.68 (0.49)	0.59 (0.1)	0.7 (0.28)	0.62 (0.55)	0.67 (0.01)
cluster_laststate	0.6 (0.37)	0.88 (0.75)	0.5 (0.01)	0.64 (0.17)	0.7 (0.62)	**0.76 (0.1)**
prefix_laststate	0.58 (0.41)	0.9 (0.79)	0.52 (0.01)	0.61 (0.15)	0.68 (0.54)	0.62 (0.01)
prefix_agg	0.59 (0.43)	0.6 (0.05)	0.54 (0.01)	0.6 (0.17)	0.6 (0.54)	0.63 (0.01)
state_agg	0.66 (0.46)	0.65 (0.46)	0.6 (0.1)	0.69 (0.27)	0.67 (0.58)	0.65 (0.01)
cluster_agg	0.61 (0.38)	0.84 (0.72)	0.54 (0.01)	0.62 (0.11)	0.63 (0.53)	0.65 (0.01)
prefix_index	0.58 (0.09)	0.66 (0.13)	0.52 (0.01)	0.54 (0.01)	0.63 (0.56)	0.69 (0.01)
single_laststate	0.57 (0.56)	0.63 (0.01)	0.51 (0.01)	0.58 (0.08)	0.68 (0.54)	0.74 (0.01)

	logit					
	bpic2015_2	traffic	sepsis_3	sepsis_2	bpic2011_1	bpic2015_3
single_agg	0.89 (0.01)	0.6 (0.51)	0.74 (0.22)	0.51 (0.01)	0.49 (0.58)	0.61 (0.11)
state_laststate	0.87 (0.37)	0.63 (0.62)	0.7 (0.08)	0.34 (0.01)	0.62 (0.68)	0.64 (0.09)
cluster_laststate	0.67 (0.33)	0.63 (0.61)	0.63 (0.01)	0.51 (0.01)	0.89 (0.73)	0.63 (0.11)
prefix_laststate	0.75 (0.01)	0.61 (0.55)	0.6 (0.01)	0.51 (0.01)	0.57 (0.61)	0.47 (0.01)
prefix_agg	0.88 (0.1)	0.64 (0.52)	0.67 (0.34)	0.51 (0.01)	0.55 (0.57)	0.61 (0.11)
state_agg	0.9 (0.49)	0.65 (0.61)	0.51 (0.01)	0.34 (0.01)	0.62 (0.68)	0.62 (0.09)
cluster_agg	0.88 (0.09)	0.66 (0.63)	0.58 (0.01)	0.51 (0.01)	0.57 (0.52)	0.46 (0.01)
prefix_index	0.69 (0.26)	0.58 (0.01)	0.6 (0.01)	0.88 (0.57)	0.54 (0.69)	0.57 (0.07)
single_laststate	0.75 (0.01)	0.6 (0.54)	0.72 (0.29)	0.51 (0.01)	0.53 (0.63)	0.46 (0.01)

	logit					
	bpic2015_1	bpic2011_2	bpic2011_3	bpic2017_a	bpic2015_5	bpic2017_r
single_agg	0.69 (0.04)	0.58 (0.78)	0.91 (0.8)	0.84 (0.68)	0.64 (0.25)	0.81 (0.37)
state_laststate	0.71 (0.45)	0.51 (0.68)	0.76 (0.57)	0.8 (0.57)	0.68 (0.34)	0.68 (0.03)
cluster_laststate	0.59 (0.3)	0.92 (0.87)	0.85 (0.01)	0.78 (0.56)	0.67 (0.37)	0.66 (0.01)
prefix_laststate	0.64 (0.24)	0.54 (0.78)	0.77 (0.06)	0.78 (0.56)	0.65 (0.23)	0.67 (0.01)
prefix_agg	0.64 (0.23)	0.58 (0.78)	0.92 (0.79)	0.8 (0.56)	0.65 (0.32)	0.69 (0.01)
state_agg	0.71 (0.42)	0.51 (0.72)	0.85 (0.74)	0.84 (0.7)	0.63 (0.29)	0.7 (0.04)
cluster_agg	0.66 (0.09)	0.62 (0.78)	0.74 (0.6)	0.84 (0.69)	0.66 (0.3)	0.68 (0.01)
prefix_index	0.52 (0.07)	0.94 (0.91)	0.92 (0.81)	0.57 (0.02)	0.67 (0.32)	0.55 (0.02)
single_laststate	0.63 (0.17)	0.62 (0.78)	0.94 (0.78)	0.83 (0.67)	0.67 (0.37)	0.66 (0.01)

	logit	
	bpic2017_c	sepsis_1
single_agg	**0.84 (0.68)**	0.5 (0.05)
state_laststate	0.73 (0.62)	0.52 (0.02)
cluster_laststate	0.72 (0.63)	0.44 (0.01)
prefix_laststate	0.71 (0.61)	0.48 (0.06)
prefix_agg	0.8 (0.56)	0.52 (0.12)
state_agg	0.73 (0.64)	0.51 (0.04)
cluster_agg	0.72 (0.63)	0.52 (0.01)
prefix_index	0.55 (0.4)	0.46 (0.01)
single_laststate	0.83 (0.67)	0.46 (0.01)

Table B.3: Overall AUC (F-score) for **RF**

	rf					
	bpic2015_2	bpic2015_4	bpic2012_c	bpic2011_4	bpic2012_d	traffic
cluster_laststate	0.7 (0.54)	0.67 (0.18)	0.69 (0.42)	0.86 (0.71)	0.61 (0.09)	0.65 (0.66)
cluster_agg	0.98 (0.95)	0.69 (0.15)	0.68 (0.41)	0.87 (0.74)	0.56 (0.21)	0.66 (0.66)
single_agg	0.98 (0.98)	0.67 (0.2)	0.69 (0.4)	0.87 (0.75)	0.59 (0.2)	0.65 (0.66)
prefix_agg	0.98 (0.96)	0.66 (0.19)	0.69 (0.4)	0.88 (0.77)	0.57 (0.18)	0.63 (0.51)
prefix_laststate	0.91 (0.24)	0.7 (0.24)	0.68 (0.4)	0.89 (0.76)	0.59 (0.17)	0.64 (0.65)
single_laststate	0.91 (0.24)	0.69 (0.21)	0.68 (0.44)	0.89 (0.74)	0.58 (0.18)	0.66 (0.66)
prefix_index	0.74 (0.49)	0.69 (0.26)	0.55 (0.24)	0.9 (0.75)	0.57 (0.06)	0.71 (0.11)
state_agg	0.89 (0.4)	0.67 (0.13)	0.7 (0.42)	0.83 (0.66)	0.59 (0.17)	0.66 (0.66)
state_laststate	0.84 (0.34)	0.69 (0.24)	0.69 (0.42)	0.83 (0.68)	0.6 (0.17)	0.65 (0.66)

	rf					
	bpic2012_a	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1
cluster_laststate	0.65 (0.63)	0.61 (0.08)	0.93 (0.74)	0.91 (0.84)	0.5 (−0.0)	0.78 (0.47)
cluster_agg	0.67 (0.65)	0.63 (0.25)	0.91 (0.74)	0.99 (0.99)	0.5 (−0.0)	0.87 (0.73)
single_agg	0.67 (0.64)	0.63 (0.19)	0.94 (0.77)	0.99 (0.87)	0.5 (−0.0)	**0.88** (**0.72**)
prefix_agg	0.68 (0.64)	0.63 (0.22)	0.9 (0.73)	0.98 (0.91)	0.5 (−0.0)	0.82 (0.65)
prefix_laststate	0.66 (0.61)	0.62 (0.24)	0.95 (0.75)	0.93 (0.88)	0.5 (−0.0)	0.74 (0.41)
single_laststate	0.65 (0.62)	0.62 (0.23)	0.94 (0.74)	0.9 (0.83)	0.5 (−0.0)	0.81 (0.53)
prefix_index	0.57 (0.45)	0.61 (0.26)	0.95 (0.81)	0.92 (0.87)	0.85 (0.68)	0.65 (0.19)
state_agg	0.68 (0.64)	0.64 (0.17)	0.88 (0.76)	0.72 (0.8)	0.33 (−0.0)	0.8 (0.64)
state_laststate	0.67 (0.62)	0.62 (0.28)	0.93 (0.81)	0.62 (0.75)	0.33 (−0.0)	0.77 (0.49)

	rf					
	bpic2015_5	sepsis_3	bpic2017_c	production	bpic2017_r	sepsis_1
cluster_laststate	0.69 (0.46)	0.72 (0.29)	0.79 (0.72)	0.64 (0.56)	0.8 (0.46)	0.39 (0.01)
cluster_agg	0.68 (0.5)	0.72 (0.29)	0.79 (0.72)	0.55 (0.48)	0.8 (0.47)	0.41 (−0.0)
single_agg	0.69 (0.41)	0.72 (0.35)	0.83 (0.67)	**0.73** (**0.61**)	0.81 (0.43)	0.36 (−0.0)
prefix_agg	0.68 (0.48)	0.72 (0.3)	0.79 (0.72)	0.7 (0.6)	0.8 (0.47)	0.48 (0.02)
prefix_laststate	0.72 (0.34)	0.74 (0.21)	0.79 (0.72)	0.68 (0.55)	0.8 (0.46)	0.46 (0.03)
single_laststate	0.67 (0.46)	0.74 (0.14)	0.82 (0.66)	0.6 (0.56)	0.8 (0.45)	0.47 (−0.0)
prefix_index	0.68 (0.46)	0.72 (0.35)	0.55 (0.47)	0.7 (0.58)	0.58 (0.02)	0.51 (−0.0)
state_agg	0.67 (0.46)	0.7 (0.08)	0.79 (0.72)	0.54 (0.44)	0.8 (0.47)	0.43 (−0.0)
state_laststate	0.72 (0.22)	0.69 (0.3)	0.79 (0.72)	0.63 (0.57)	0.81 (0.46)	0.46 (−0.0)

	rf	
	bpic2011_1	bpic2017_a
cluster_laststate	0.86 (0.73)	0.83 (0.7)
cluster_agg	0.93 (0.84)	0.83 (0.68)
single_agg	**0.94** (**0.87**)	0.83 (0.67)
prefix_agg	**0.94** (**0.88**)	0.83 (0.71)
prefix_laststate	0.89 (0.75)	0.77 (0.55)
single_laststate	0.82 (0.7)	0.82 (0.66)
prefix_index	0.85 (0.73)	0.59 (0.39)
state_agg	0.92 (0.85)	0.83 (0.69)
state_laststate	0.86 (0.72)	0.79 (0.56)

Table B.4: Overall AUC (F-score) for **SVM**

	svm					
	bpic2012_a	bpic2011_4	bpic2012_d	bpic2012_c	production	bpic2015_4
single_agg	0.51 (0.72)	0.51 (0.01)	0.51 (0.01)	0.51 (0.01)	0.62 (0.54)	0.51 (0.01)
state_laststate	0.62 (0.34)	0.5 (0.1)	0.53 (0.1)	0.6 (0.14)	0.62 (0.56)	0.54 (0.01)
cluster_laststate	0.59 (0.47)	0.76 (0.01)	0.52 (0.01)	0.59 (0.01)	0.58 (0.12)	0.51 (0.01)
prefix_laststate	0.51 (0.31)	0.55 (0.01)	0.51 (0.01)	0.52 (0.01)	0.64 (0.54)	0.51 (0.01)
prefix_agg	0.51 (0.31)	0.53 (0.01)	0.51 (0.01)	0.51 (0.01)	0.51 (0.01)	0.5 (0.01)
state_agg	0.61 (0.37)	0.5 (0.01)	0.53 (0.1)	0.61 (0.14)	0.67 (0.39)	0.54 (0.01)
cluster_agg	0.49 (0.01)	0.55 (0.01)	0.54 (0.01)	0.59 (0.01)	0.59 (0.31)	0.51 (0.01)
prefix_index	0.51 (0.01)	0.51 (0.01)	0.51 (0.01)	0.51 (0.01)	0.53 (0.01)	0.55 (0.01)
single_laststate	0.53 (0.72)	0.61 (0.01)	0.51 (0.01)	0.51 (0.01)	0.66 (0.57)	0.51 (0.01)

	svm					
	bpic2015_2	traffic	sepsis_3	sepsis_2	bpic2011_1	bpic2015_3
single_agg	0.51 (0.01)	0.6 (0.51)	0.55 (0.01)	0.51 (0.01)	0.56 (0.1)	0.54 (0.08)
state_laststate	0.72 (0.31)	0.64 (0.54)	0.6 (0.13)	0.34 (0.01)	0.62 (0.06)	0.55 (0.04)
cluster_laststate	0.75 (0.58)	0.66 (0.67)	0.72 (0.01)	0.51 (0.01)	0.5 (0.34)	0.56 (0.01)
prefix_laststate	0.56 (0.01)	0.66 (0.67)	0.62 (0.01)	0.51 (0.01)	0.52 (0.01)	0.57 (0.11)
prefix_agg	0.51 (0.01)	0.52 (0.28)	0.65 (0.01)	0.51 (0.01)	0.52 (0.46)	0.55 (0.01)
state_agg	0.71 (0.31)	0.63 (0.45)	0.53 (0.01)	0.34 (0.01)	0.62 (0.06)	0.5 (0.01)
cluster_agg	0.51 (0.01)	0.67 (0.67)	0.62 (0.28)	0.51 (0.01)	0.5 (0.01)	0.5 (0.01)
prefix_index	0.61 (0.01)	0.51 (0.01)	0.51 (0.01)	0.88 (0.01)	0.51 (0.01)	0.51 (0.01)
single_laststate	0.51 (0.01)	0.6 (0.54)	0.69 (0.01)	0.51 (0.01)	0.51 (0.01)	0.51 (0.01)

	svm					
	bpic2015_1	bpic2011_2	bpic2011_3	bpic2017_a	bpic2015_5	bpic2017_r
single_agg	0.51 (0.01)	0.51 (0.78)	0.54 (0.01)	0.84 (0.68)	0.51 (0.01)	0.51 (0.01)
state_laststate	0.62 (0.34)	0.56 (0.64)	0.59 (0.01)	0.53 (0.12)	0.68 (0.01)	0.51 (0.01)
cluster_laststate	0.47 (0.01)	0.51 (0.78)	0.83 (0.01)	0.46 (0.01)	0.51 (0.01)	0.51 (0.01)
prefix_laststate	0.51 (0.01)	0.5 (0.78)	0.61 (0.01)	0.51 (0.01)	0.54 (0.01)	0.51 (0.01)
prefix_agg	0.51 (0.01)	0.51 (0.78)	0.51 (0.01)	0.51 (0.01)	0.65 (0.01)	0.51 (0.01)
state_agg	0.63 (0.34)	0.56 (0.64)	0.69 (0.56)	0.53 (0.12)	0.51 (0.01)	0.54 (0.01)
cluster_agg	0.65 (0.01)	0.52 (0.78)	0.7 (0.01)	0.46 (0.01)	0.62 (0.29)	0.54 (0.01)
prefix_index	0.51 (0.01)	0.67 (0.81)	0.51 (0.01)	0.51 (0.01)	0.51 (0.01)	0.54 (0.02)
single_laststate	0.53 (0.01)	0.51 (0.01)	0.6 (0.01)	0.83 (0.67)	0.63 (0.01)	0.51 (0.01)

	svm	
	bpic2017_c	sepsis_1
single_agg	**0.84** (**0.68**)	0.5 (0.01)
state_laststate	0.8 (0.57)	0.51 (0.01)
cluster_laststate	0.51 (0.01)	0.51 (0.09)
prefix_laststate	0.51 (0.01)	0.51 (0.01)
prefix_agg	0.51 (0.01)	0.53 (0.01)
state_agg	**0.84** (**0.7**)	0.51 (0.01)
cluster_agg	0.54 (0.31)	0.54 (0.04)
prefix_index	0.51 (0.1)	0.47 (0.01)
single_laststate	0.83 (0.67)	0.51 (0.01)

Table B.5: Execution time for **XGBoost**

	bpic2015_2		bpic2012_C		bpic2015_4	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	**155.15 ± 33.76**	88.95 ± 0.05	128.65 ± 46.04	20.76 ± 0.01	350.22 ± 35.74	52.66 ± 0.03
cluster_agg	10339.59 ± 48.73	37.31 ± 0.04	90667.34 ± 39.43	**10.97 ± 0.01**	6946.37 ± 38.25	38.83 ± 0.04
single_agg	98.28 ± 28.78	153.25 ± 0.06	383.15 ± 38.49	23.25 ± 0.01	750.38 ± 21.01	156.81 ± 0.06
prefix_agg	9708.46 ± 40.16	**18.57 ± 46.08**	4903.78 ± 34.28	32.99 ± 40.37	7482.45 ± 39.9	**34.09 ± 24.41**
prefix_laststate	754.71 ± 36.32	48.66 ± 40.01	953.5 ± 43.29	43.86 ± 48.24	497.23 ± 22.65	57.05 ± 46.5
single_laststate	319.07 ± 38.04	220.7 ± 0.06	525.63 ± 46.1	27.49 ± 0.01	960.65 ± 43.82	228.46 ± 0.06
prefix_index	430.29 ± 44.81	33.02 ± 30.72	543.77 ± 49.25	43.96 ± 30.92	252.55 ± 33.16	34.32 ± 47.61
state_agg	400.14 ± 42.58	66.7 ± 0.02	25.95 ± 29.97	32.52 ± 0.01	**140.42 ± 35.49**	67.15 ± 0.02
state_laststate	180.39 ± 21.33	261.54 ± 0.07	**20.36 ± 25.85**	32.51 ± 0.01	140.95 ± 48.11	262.43 ± 0.07

	bpic2012_D		bpic2017_A		bpic2011_4	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	143.99 ± 38.24	20.73 ± 0.01	1661.58 ± 35.09	42.15 ± 0.02	380.24 ± 40.73	215.48 ± 0.12
cluster_agg	7600.58 ± 24.58	**10.93 ± 0.01**	481255.91 ± 22.52	27.61 ± 0.02	85945.23 ± 37.36	160.7 ± 0.18
single_agg	1211.48 ± 45.89	51.29 ± 0.02	5338.84 ± 36.39	77.46 ± 0.02	213.39 ± 35.89	192.4 ± 0.08
prefix_agg	4976.51 ± 26.69	33.99 ± 45.45	26745.19 ± 360.54	32.34 ± 36.73	4203.23 ± 39.89	**29.42 ± 28.8**
prefix_laststate	1006.64 ± 33.05	39.31 ± 35.8	2773.21 ± 40.39	46.05 ± 30.28	1372.81 ± 49.64	33.64 ± 41.34
single_laststate	688.33 ± 27.89	27.68 ± 0.01	7068.01 ± 27.19	75.98 ± 0.02	557.07 ± 27.44	324.08 ± 0.1
prefix_index	565.01 ± 23.71	46.58 ± 36.85	3646.22 ± 25.15	**21.56 ± 46.55**	120.55 ± 46.8	33.0 ± 45.63
state_agg	39.75 ± 30.98	32.51 ± 0.01	**178.05 ± 44.89**	69.55 ± 0.02	180.46 ± 40.44	571.03 ± 0.18
state_laststate	**32.53 ± 49.35**	32.82 ± 0.01	492.61 ± 23.67	73.64 ± 0.02	**114.6 ± 44.6**	169.04 ± 0.05

	traffic		bpic2012_A		bpic2015_3	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	2626.8 ± 46.33	126.59 ± 0.02	318.36 ± 34.44	21.13 ± 0.01	92.11 ± 35.75	80.09 ± 0.04
cluster_agg	4796.83 ± 41.67	67.17 ± 0.03	43748.93 ± 46.71	**10.18 ± 0.01**	31157.59 ± 30.3	43.63 ± 0.05
single_agg	4390.07 ± 31.27	172.63 ± 0.02	167.15 ± 43.52	22.56 ± 0.01	251.5 ± 23.78	178.36 ± 0.07
prefix_agg	15334.24 ± 46.22	**18.85 ± 23.09**	5041.63 ± 39.22	35.15 ± 48.92	15363.77 ± 49.84	**21.53 ± 41.61**
prefix_laststate	2909.98 ± 32.16	32.33 ± 31.25	941.92 ± 30.72	28.35 ± 44.79	1115.09 ± 41.82	36.07 ± 20.77
single_laststate	3191.65 ± 36.36	213.8 ± 0.02	502.66 ± 30.21	27.52 ± 0.01	399.74 ± 48.33	259.21 ± 0.07
prefix_index	13999.84 ± 39.84	49.96 ± 21.63	549.23 ± 22.77	41.12 ± 35.42	479.99 ± 43.74	25.35 ± 38.09
state_agg	355.86 ± 24.19	228.84 ± 0.03	420.43 ± 21.35	32.02 ± 0.01	**90.35 ± 45.17**	78.01 ± 0.02
state_laststate	**299.2 ± 38.47**	240.54 ± 0.02	**310.29 ± 41.66**	31.69 ± 0.01	270.17 ± 39.35	304.13 ± 0.08

	bpic2011_3		bpic2011_2		sepsis_2	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	170.76 ± 29.38	350.48 ± 0.21	183.78 ± 33.1	224.69 ± 0.12	13.46 ± 41.73	134.97 ± 0.05
cluster_agg	715.35 ± 48.11	284.84 ± 0.41	154836.61 ± 23.94	151.75 ± 0.15	1918.45 ± 23.92	44.16 ± 0.03
single_agg	88.68 ± 29.3	182.71 ± 0.06	182.15 ± 30.42	165.74 ± 0.06	31.19 ± 25.02	234.25 ± 0.06
prefix_agg	1866.84 ± 28.72	33.27 ± 25.24	3427.02 ± 29.52	**23.56 ± 26.01**	469.06 ± 35.17	28.86 ± 22.68
prefix_laststate	799.55 ± 33.41	47.78 ± 38.9	1531.87 ± 30.04	46.99 ± 34.6	95.84 ± 42.61	61.01 ± 45.21
single_laststate	162.19 ± 45.65	260.21 ± 0.07	218.96 ± 40.22	225.38 ± 0.07	15.32 ± 36.97	260.7 ± 0.05
prefix_index	124.38 ± 36.32	**25.32 ± 37.02**	128.14 ± 47.44	48.25 ± 37.23	103.17 ± 43.57	**27.99 ± 22.88**
state_agg	20.54 ± 33.67	690.15 ± 0.18	49.37 ± 45.15	705.03 ± 0.23	**2.5 ± 23.83**	230.44 ± 0.04
state_laststate	**11.98 ± 47.79**	239.88 ± 0.06	**43.65 ± 40.7**	219.83 ± 0.06	7.34 ± 34.69	509.39 ± 0.14

	bpic2015_1		bpic2015_5		sepsis_3	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	78.06 ± 41.78	115.47 ± 0.06	54.67 ± 25.53	63.3 ± 0.03	42.27 ± 32.76	117.48 ± 0.04
cluster_agg	757.46 ± 31.44	38.92 ± 0.04	27222.9 ± 21.7	73.49 ± 0.09	4695.72 ± 35.17	37.71 ± 0.03
single_agg	120.09 ± 37.24	191.21 ± 0.07	122.79 ± 21.47	212.25 ± 0.08	25.36 ± 32.85	297.98 ± 0.09
prefix_agg	9260.39 ± 25.68	27.28 ± 37.79	11399.57 ± 47.88	39.73 ± 22.06	1439.04 ± 37.96	**20.91 ± 24.07**
prefix_laststate	528.12 ± 38.55	57.82 ± 48.03	950.31 ± 45.69	60.6 ± 41.29	274.36 ± 37.15	61.83 ± 20.55
single_laststate	256.8 ± 47.91	278.54 ± 0.08	564.32 ± 40.98	281.73 ± 0.08	38.85 ± 32.04	462.45 ± 0.14
prefix_index	296.75 ± 39.56	**26.22 ± 30.42**	469.52 ± 45.23	**38.82 ± 26.4**	105.83 ± 35.57	43.93 ± 21.89
state_agg	**40.98 ± 29.46**	82.71 ± 0.03	**6.35 ± 21.47**	83.26 ± 0.02	**4.29 ± 38.78**	29.23 ± 0.04
state_laststate	200.79 ± 25.68	320.48 ± 0.09	29.47 ± 46.33	350.1 ± 0.09	4.85 ± 43.13	164.35 ± 0.03

	bpic2017_C		production		sepsis_1	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	10772.93 ± 380.21	68.79 ± 0.03	1750.72 ± 33.97	202.97 ± 0.06	29.17 ± 45.05	102.13 ± 0.04
cluster_agg	1144.62 ± 49.96	36.6 ± 0.03	3696.21 ± 38.05	36.51 ± 0.02	5021.58 ± 44.56	40.06 ± 0.03
single_agg	1759.09 ± 29.79	59.73 ± 0.02	**12.06 ± 47.58**	227.73 ± 0.05	25.94 ± 38.1	211.18 ± 0.06
prefix_agg	14694.85 ± 41.63	27.99 ± 37.85	533.42 ± 29.4	**22.08 ± 31.24**	1404.91 ± 41.42	**32.95 ± 25.7**
prefix_laststate	2816.38 ± 36.4	49.13 ± 35.63	95.69 ± 40.43	49.92 ± 26.0	263.29 ± 39.48	54.56 ± 42.06
single_laststate	9308.12 ± 23.68	74.23 ± 0.02	42.43 ± 38.88	141.01 ± 0.02	74.45 ± 42.26	193.73 ± 0.04
prefix_index	3660.0 ± 23.11	**21.89 ± 44.53**	25.01 ± 44.03	48.6 ± 29.28	99.36 ± 40.49	43.4 ± 33.26
state_agg	**189.17 ± 37.82**	72.5 ± 0.02	228.91 ± 49.88	349.99 ± 0.07	**2.74 ± 34.41**	184.66 ± 0.04
state_laststate	246.39 ± 22.8	74.08 ± 0.02	26.02 ± 38.05	159.1 ± 0.02	11.48 ± 21.42	488.28 ± 0.1

	bpic2011_1		bpic2017_R	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	335.7 ± 23.74	244.77 ± 0.12	975.59 ± 31.27	42.15 ± 0.02
cluster_agg	5647.68 ± 35.97	410.53 ± 0.52	1095692.43 ± 34.56	28.11 ± 0.02
single_agg	125.74 ± 27.11	182.61 ± 0.06	1447.26 ± 44.14	75.81 ± 0.02
prefix_agg	2506.78 ± 38.31	37.66 ± 24.28	14701.27 ± 38.67	**21.46 ± 47.35**
prefix_laststate	980.86 ± 46.41	50.48 ± 22.7	2801.71 ± 30.13	36.45 ± 30.22
single_laststate	176.97 ± 24.95	239.7 ± 0.07	4163.11 ± 26.54	74.8 ± 0.02
prefix_index	116.88 ± 33.74	**29.48 ± 43.14**	3766.01 ± 38.77	34.55 ± 37.58
state_agg	510.69 ± 39.97	716.8 ± 0.21	**112.68 ± 27.15**	69.11 ± 0.02
state_laststate	**170.12 ± 45.36**	226.51 ± 0.05	159.32 ± 42.52	73.81 ± 0.02

Table B.6: Execution time for **Logit**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	230.13 ± 26.98	75.08 ± 0.04	230.23 ± 34.37	21.13 ± 0.01	210.97 ± 26.62	79.53 ± 0.04
cluster_agg	5964.06 ± 24.53	52.3 ± 0.06	4700.15 ± 26.52	**10.9 ± 0.01**	4417.1 ± 31.66	38.84 ± 0.04
single_agg	310.35 ± 45.84	177.9 ± 0.07	150.57 ± 39.46	22.75 ± 0.01	160.78 ± 29.67	176.45 ± 0.07
prefix_agg	9518.33 ± 35.43	**26.31 ± 30.23**	5044.79 ± 25.04	39.68 ± 28.84	6874.94 ± 23.32	33.54 ± 36.82
prefix_laststate	551.22 ± 32.56	40.05 ± 38.43	951.46 ± 29.73	54.46 ± 30.56	443.72 ± 39.73	28.25 ± 48.03
single_laststate	220.7 ± 46.83	65.27 ± 0.02	310.38 ± 49.46	27.83 ± 0.01	110.4 ± 37.7	115.78 ± 0.04
prefix_index	310.81 ± 26.03	31.9 ± 36.63	551.75 ± 21.38	37.76 ± 35.04	256.32 ± 23.69	**23.47 ± 42.01**
state_agg	**210.7 ± 44.5**	138.25 ± 0.04	**140.35 ± 32.54**	32.19 ± 0.01	**97.9 ± 25.86**	79.8 ± 0.02
state_laststate	230.04 ± 42.21	328.53 ± 0.1	600.87 ± 43.38	32.18 ± 0.01	140.2 ± 34.22	268.74 ± 0.07

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	170.31 ± 33.85	21.26 ± 0.01	910.42 ± 36.13	41.42 ± 0.02	83.13 ± 42.35	226.8 ± 0.12
cluster_agg	5737.0 ± 39.49	**11.09 ± 0.01**	72707.65 ± 47.02	28.32 ± 0.02	20846.43 ± 48.3	87.95 ± 0.09
single_agg	90.5 ± 42.39	23.6 ± 0.01	2790.28 ± 30.39	78.53 ± 0.02	46.28 ± 42.52	307.56 ± 0.13
prefix_agg	5089.26 ± 48.04	20.5 ± 24.21	14728.8 ± 38.32	**21.42 ± 29.34**	3543.28 ± 26.77	**17.98 ± 35.73**
prefix_laststate	950.96 ± 20.86	50.3 ± 48.43	2845.97 ± 21.62	34.77 ± 22.19	1575.03 ± 34.85	50.5 ± 31.82
single_laststate	180.8 ± 24.81	27.46 ± 0.01	308.8 ± 26.59	94.4 ± 0.02	15.9 ± 30.47	163.64 ± 0.05
prefix_index	553.82 ± 29.45	39.65 ± 47.02	3658.44 ± 29.3	37.79 ± 33.73	122.41 ± 36.46	43.74 ± 45.33
state_agg	**40.92 ± 42.96**	33.46 ± 0.01	320.78 ± 24.34	89.48 ± 0.02	16.85 ± 23.27	525.57 ± 0.16
state_laststate	80.61 ± 48.75	33.38 ± 0.01	**240.61 ± 46.42**	73.18 ± 0.01	**6.0 ± 49.77**	147.87 ± 0.04

	traffic		**bpic2012_A**		**bpic2015_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	720.76 ± 26.24	120.79 ± 0.02	280.66 ± 22.67	20.78 ± 0.01	61.04 ± 21.01	95.61 ± 0.05
cluster_agg	14168.79 ± 38.73	65.68 ± 0.02	5982.58 ± 26.93	**10.92 ± 0.01**	12751.53 ± 49.04	38.29 ± 0.04
single_agg	580.59 ± 48.87	168.51 ± 0.02	**150.81 ± 37.34**	22.7 ± 0.01	830.31 ± 40.89	219.52 ± 0.09
prefix_agg	15429.4 ± 35.88	**30.8 ± 44.84**	4915.05 ± 29.73	36.81 ± 24.43	14189.44 ± 25.02	**23.78 ± 20.5**
prefix_laststate	2964.22 ± 33.96	66.43 ± 34.69	931.19 ± 40.92	55.77 ± 39.17	942.02 ± 35.75	65.17 ± 44.94
single_laststate	570.36 ± 31.22	203.86 ± 0.02	310.87 ± 35.46	28.26 ± 0.01	160.44 ± 43.88	79.68 ± 0.02
prefix_index	14002.19 ± 40.09	39.56 ± 45.1	555.39 ± 38.19	24.78 ± 23.66	484.18 ± 47.32	41.55 ± 28.7
state_agg	**490.43 ± 48.91**	222.94 ± 0.02	820.96 ± 25.99	32.48 ± 0.01	**130.11 ± 44.25**	217.23 ± 0.05
state_laststate	1003.56 ± 22.3	237.17 ± 0.02	860.68 ± 41.27	32.66 ± 0.01	430.58 ± 42.05	385.46 ± 0.11

	bpic2011_3		**bpic2011_2**		**sepsis_2**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	26.47 ± 34.78	254.85 ± 0.13	28.93 ± 33.99	109.78 ± 0.06	7.75 ± 22.94	128.17 ± 0.04
cluster_agg	10664.12 ± 48.85	217.42 ± 0.26	10913.99 ± 20.98	71.6 ± 0.05	755.71 ± 35.99	43.59 ± 0.03
single_agg	15.65 ± 31.63	240.43 ± 0.08	28.88 ± 33.35	215.71 ± 0.09	8.47 ± 46.58	311.83 ± 0.08
prefix_agg	1651.42 ± 43.94	**35.6 ± 49.01**	3453.69 ± 32.31	**24.09 ± 26.01**	477.24 ± 35.57	38.94 ± 24.66
prefix_laststate	858.23 ± 23.45	64.0 ± 28.91	1518.31 ± 36.97	46.04 ± 41.7	89.15 ± 42.26	**29.52 ± 33.24**
single_laststate	17.47 ± 42.0	236.72 ± 0.06	33.2 ± 28.87	248.35 ± 0.07	7.33 ± 30.88	252.28 ± 0.05
prefix_index	114.29 ± 44.66	45.71 ± 42.24	129.6 ± 40.99	48.12 ± 44.53	110.12 ± 36.16	46.34 ± 28.2
state_agg	10.58 ± 26.88	575.63 ± 0.15	19.5 ± 28.96	601.74 ± 0.19	**3.41 ± 42.5**	508.56 ± 0.13
state_laststate	**4.14 ± 43.85**	183.47 ± 0.04	**8.94 ± 49.37**	187.06 ± 0.05	5.38 ± 28.82	582.27 ± 0.12

	bpic2015_1		**bpic2015_5**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	26.65 ± 25.48	78.99 ± 0.04	34.15 ± 33.11	66.57 ± 0.03	11.93 ± 40.01	102.02 ± 0.04
cluster_agg	6099.77 ± 41.49	38.22 ± 0.04	14592.77 ± 28.0	60.73 ± 0.06	1526.51 ± 32.2	41.24 ± 0.03
single_agg	29.91 ± 44.36	227.85 ± 0.09	28.45 ± 28.22	189.45 ± 0.07	11.38 ± 23.22	409.91 ± 0.12
prefix_agg	7810.05 ± 22.88	35.53 ± 48.67	11829.8 ± 31.26	**21.91 ± 21.71**	1418.93 ± 20.7	**36.62 ± 43.58**
prefix_laststate	528.4 ± 30.7	59.87 ± 49.67	722.39 ± 45.02	57.25 ± 46.15	269.42 ± 25.77	66.27 ± 43.35
single_laststate	11.53 ± 47.57	84.5 ± 0.02	25.4 ± 32.41	82.21 ± 0.02	15.63 ± 30.18	187.44 ± 0.04
prefix_index	430.89 ± 41.93	**30.11 ± 31.07**	615.64 ± 31.21	35.82 ± 33.33	101.8 ± 34.91	39.46 ± 30.43
state_agg	**4.28 ± 47.57**	102.48 ± 0.03	**3.19 ± 26.0**	82.13 ± 0.02	**1.55 ± 49.85**	200.22 ± 0.04
state_laststate	32.99 ± 37.4	401.59 ± 0.12	23.59 ± 23.09	326.43 ± 0.09	3.26 ± 26.24	231.22 ± 0.04

	bpic2017_C		**production**		**sepsis_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	960.63 ± 42.57	41.75 ± 0.02	12.35 ± 25.22	108.67 ± 0.03	15.49 ± 41.63	116.92 ± 0.05
cluster_agg	674260.87 ± 39.67	**24.32 ± 0.02**	3112.47 ± 34.2	39.29 ± 0.03	2177.16 ± 48.63	35.5 ± 0.03
single_agg	1759.09 ± 30.26	59.73 ± 0.02	2.65 ± 28.84	110.26 ± 0.02	11.23 ± 26.47	317.06 ± 0.09
prefix_agg	14694.85 ± 29.07	28.05 ± 49.27	1493.41 ± 46.73	**17.87 ± 41.06**	1404.2 ± 38.75	28.19 ± 32.49
prefix_laststate	2852.87 ± 20.75	42.29 ± 41.6	202.06 ± 33.7	58.7 ± 45.8	272.99 ± 21.33	57.92 ± 40.56
single_laststate	9308.12 ± 47.21	74.23 ± 0.02	5.28 ± 37.88	162.02 ± 0.03	13.9 ± 29.8	160.23 ± 0.04
prefix_index	3640.72 ± 35.32	48.61 ± 28.69	23.72 ± 24.47	43.04 ± 26.37	127.17 ± 38.88	**22.6 ± 41.33**
state_agg	**150.67 ± 25.2**	69.94 ± 0.02	**2.23 ± 43.38**	255.01 ± 0.05	**5.48 ± 20.92**	492.22 ± 0.11
state_laststate	220.9 ± 37.37	73.47 ± 0.01	2.83 ± 47.74	190.27 ± 0.04	9.31 ± 31.44	393.46 ± 0.09

	bpic2011_1		**bpic2017_R**			
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)		
cluster_laststate	19.98 ± 45.34	109.98 ± 0.06	980.51 ± 32.82	43.29 ± 0.02		
cluster_agg	13445.93 ± 37.66	191.01 ± 0.22	28605.61 ± 30.76	**21.21 ± 0.02**		
single_agg	18.35 ± 46.45	203.34 ± 0.07	346.81 ± 46.56	77.54 ± 0.02		
prefix_agg	2352.37 ± 43.33	**20.76 ± 49.9**	14907.29 ± 35.33	34.18 ± 33.91		
prefix_laststate	1121.32 ± 44.56	37.41 ± 35.15	2824.85 ± 46.33	34.2 ± 20.77		
single_laststate	27.65 ± 40.6	369.0 ± 0.1	171.95 ± 27.77	74.76 ± 0.02		
prefix_index	115.91 ± 47.11	38.75 ± 37.32	3643.86 ± 44.78	30.26 ± 45.78		
state_agg	16.6 ± 35.18	676.0 ± 0.2	**167.51 ± 36.16**	70.48 ± 0.02		
state_laststate	**10.91 ± 42.74**	246.41 ± 0.09	250.15 ± 24.98	72.83 ± 0.01		

Table B.7: Execution time for **RF**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	79.39 ± 29.9	247.83 ± 0.22	240.72 ± 40.84	63.01 ± 0.03	38.21 ± 47.74	193.22 ± 0.11
cluster_agg	28650.9 ± 33.92	204.83 ± 0.22	9635.2 ± 36.44	31.47 ± 0.03	9479.18 ± 26.0	139.07 ± 0.15
single_agg	42.69 ± 40.29	379.88 ± 0.15	54.64 ± 39.64	80.79 ± 0.03	320.72 ± 34.62	345.15 ± 0.13
prefix_agg	8380.55 ± 41.7	**38.16 ± 49.73**	4790.49 ± 28.71	**30.52 ± 28.05**	6042.06 ± 34.88	**27.7 ± 36.96**
prefix_laststate	1633.45 ± 21.65	42.69 ± 20.74	933.1 ± 42.66	47.98 ± 20.93	1157.74 ± 38.75	60.72 ± 25.98
single_laststate	70.87 ± 38.59	477.03 ± 0.14	80.26 ± 33.29	165.59 ± 0.05	580.58 ± 27.88	483.87 ± 0.14
prefix_index	279.65 ± 35.7	47.83 ± 34.91	564.66 ± 28.46	34.8 ± 21.42	200.26 ± 24.72	28.01 ± 36.95
state_agg	**25.99 ± 33.46**	137.99 ± 0.04	**150.07 ± 35.05**	100.2 ± 0.03	**167.45 ± 32.3**	142.81 ± 0.04
state_laststate	145.61 ± 42.63	347.85 ± 0.09	360.98 ± 45.4	140.81 ± 0.04	870.05 ± 42.65	228.63 ± 0.06

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	440.43 ± 25.72	88.7 ± 0.05	940.0 ± 36.08	100.98 ± 0.04	105.01 ± 23.99	301.7 ± 0.17
cluster_agg	38901.03 ± 31.21	41.18 ± 0.05	72707.65 ± 26.54	28.32 ± 0.02	6385.71 ± 36.34	70.42 ± 0.07
single_agg	280.08 ± 33.64	79.96 ± 0.03	279.28 ± 38.73	78.53 ± 0.02	95.05 ± 42.54	898.09 ± 0.35
prefix_agg	4982.55 ± 28.8	**19.94 ± 41.88**	14692.95 ± 26.26	**20.56 ± 21.68**	3548.16 ± 30.78	39.38 ± 49.62
prefix_laststate	970.41 ± 24.53	39.81 ± 48.44	2845.97 ± 21.39	36.64 ± 35.56	1154.36 ± 22.57	**28.26 ± 27.82**
single_laststate	398.8 ± 37.42	161.58 ± 0.05	208.8 ± 33.94	94.4 ± 0.02	**71.15 ± 39.12**	322.22 ± 0.09
prefix_index	603.56 ± 32.98	33.08 ± 27.28	3638.75 ± 43.98	37.36 ± 40.54	127.82 ± 23.58	38.24 ± 27.27
state_agg	**119.12 ± 26.38**	98.52 ± 0.03	320.78 ± 35.41	89.48 ± 0.02	155.66 ± 43.7	813.44 ± 0.25
state_laststate	360.54 ± 41.06	144.85 ± 0.05	**240.61 ± 29.93**	73.18 ± 0.01	199.5 ± 33.27	842.75 ± 0.23

	traffic		**bpic2012_A**		**bpic2015_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	720.76 ± 45.0	120.79 ± 0.02	260.97 ± 37.85	61.79 ± 0.03	346.56 ± 39.23	326.57 ± 0.34
cluster_agg	14168.79 ± 21.74	65.68 ± 0.02	11037.66 ± 39.12	**30.62 ± 0.03**	52368.78 ± 39.92	241.93 ± 0.25
single_agg	480.59 ± 33.14	168.51 ± 0.02	520.16 ± 22.8	81.78 ± 0.03	74.37 ± 24.54	364.23 ± 0.14
prefix_agg	15429.4 ± 38.12	34.79 ± 39.55	5111.24 ± 24.77	38.83 ± 21.39	13142.41 ± 49.03	**32.79 ± 49.13**
prefix_laststate	2964.22 ± 44.69	**29.45 ± 35.75**	950.04 ± 48.47	36.15 ± 47.89	2752.42 ± 39.48	36.01 ± 20.66
single_laststate	570.36 ± 49.28	203.86 ± 0.02	460.05 ± 48.79	102.33 ± 0.03	88.16 ± 35.88	376.73 ± 0.11
prefix_index	14002.19 ± 38.98	40.79 ± 47.32	550.25 ± 36.5	47.42 ± 43.18	473.24 ± 47.08	42.37 ± 27.21
state_agg	**390.43 ± 23.45**	222.94 ± 0.02	**210.9 ± 38.09**	98.25 ± 0.03	**50.4 ± 48.22**	145.39 ± 0.05
state_laststate	1003.56 ± 23.17	237.17 ± 0.02	304.77 ± 37.87	132.14 ± 0.04	126.7 ± 48.01	353.37 ± 0.12

	bpic2011_3		**bpic2011_2**		**sepsis_2**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	**21.75 ± 42.32**	455.48 ± 0.24	81.44 ± 48.5	370.33 ± 0.2	13.24 ± 39.58	182.79 ± 0.06
cluster_agg	4644.11 ± 37.85	84.71 ± 0.08	8563.93 ± 24.92	87.9 ± 0.09	7358.41 ± 35.06	123.6 ± 0.08
single_agg	37.34 ± 23.45	1222.06 ± 0.41	360.19 ± 21.65	863.46 ± 0.33	37.75 ± 40.92	833.49 ± 0.19
prefix_agg	1761.67 ± 45.35	**38.57 ± 41.26**	3407.44 ± 37.11	37.22 ± 33.41	467.38 ± 26.87	**37.66 ± 31.27**
prefix_laststate	550.21 ± 37.94	63.6 ± 39.83	1084.18 ± 45.34	**28.75 ± 45.72**	88.06 ± 23.77	65.44 ± 30.06
single_laststate	29.87 ± 36.02	398.73 ± 0.1	**200.12 ± 44.61**	259.72 ± 0.07	36.41 ± 47.71	1015.36 ± 0.19
prefix_index	127.9 ± 23.72	44.3 ± 32.18	227.31 ± 44.58	36.57 ± 33.56	107.95 ± 44.39	39.34 ± 31.59
state_agg	43.62 ± 32.44	748.18 ± 0.2	390.03 ± 23.91	660.65 ± 0.2	**7.15 ± 28.99**	558.48 ± 0.16
state_laststate	118.7 ± 49.25	959.98 ± 0.23	402 ± 39.58	704.61 ± 0.19	12.41 ± 37.41	5124.432.31 ± 0.08

	bpic2015_1		**bpic2015_5**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	47.89 ± 25.29	276.69 ± 0.15	72.99 ± 42.99	219.56 ± 0.12	21.44 ± 24.36	221.75 ± 0.08
cluster_agg	37119.26 ± 41.12	304.16 ± 0.31	39998.14 ± 36.75	211.78 ± 0.23	8844.98 ± 37.04	106.38 ± 0.08
single_agg	30.53 ± 26.18	373.1 ± 0.15	68.64 ± 28.47	407.84 ± 0.16	32.79 ± 24.12	560.62 ± 0.16
prefix_agg	7840.51 ± 44.23	**38.16 ± 25.69**	11215.78 ± 46.01	31.82 ± 21.88	1476.15 ± 29.78	**32.41 ± 43.51**
prefix_laststate	1869.82 ± 39.33	48.76 ± 35.28	2115.03 ± 45.55	41.21 ± 47.31	265.49 ± 32.24	51.79 ± 49.83
single_laststate	54.61 ± 23.3	567.44 ± 0.16	142.24 ± 28.49	521.64 ± 0.15	34.64 ± 45.25	1101.64 ± 0.25
prefix_index	255.78 ± 48.66	43.24 ± 38.72	383.89 ± 23.18	**24.81 ± 26.89**	105.34 ± 31.0	37.16 ± 41.9
state_agg	**29.92 ± 36.21**	156.98 ± 0.05	**53.21 ± 43.05**	149.78 ± 0.05	**11.14 ± 46.38**	510.09 ± 0.08
state_laststate	143.59 ± 47.14	337.18 ± 0.1	298.03 ± 33.35	424.24 ± 0.11	21.97 ± 36.87	702.25 ± 0.14

	bpic2017_C		**production**		**sepsis_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	209.97 ± 20.92	136.92 ± 0.06	27.84 ± 35.74	225.37 ± 0.07	30.46 ± 21.23	246.43 ± 0.09
cluster_agg	674260.87 ± 40.99	24.32 ± 0.02	14772.24 ± 39.47	119.96 ± 0.07	3388.72 ± 45.33	70.85 ± 0.05
single_agg	1759.09 ± 39.52	59.73 ± 0.02	**3.18 ± 48.38**	263.61 ± 0.05	**6.09 ± 27.71**	328.73 ± 0.09
prefix_agg	14694.85 ± 27.9	**22.8 ± 42.44**	727.43 ± 35.21	**30.01 ± 47.32**	1386.85 ± 37.85	**27.94 ± 36.83**
prefix_laststate	2876.22 ± 25.09	61.2 ± 39.07	217.27 ± 23.55	62.36 ± 32.28	272.2 ± 48.25	60.74 ± 26.43
single_laststate	9308.12 ± 44.39	74.23 ± 0.02	11.3 ± 29.74	330.46 ± 0.05	51.64 ± 41.82	1017.07 ± 0.23
prefix_index	3678.15 ± 36.44	49.54 ± 44.11	24.19 ± 33.97	33.51 ± 41.48	115.89 ± 25.04	32.82 ± 41.37
state_agg	**205.67 ± 28.4**	69.94 ± 0.02	11.42 ± 45.08	1090.07 ± 0.21	15.5 ± 26.56	941.38 ± 0.23
state_laststate	440.99 ± 29.94	188.1 ± 0.04	26.19 ± 49.05	673.87 ± 0.12	6.95 ± 37.4	287.81 ± 0.07

	bpic2011_1		**bpic2017_R**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	93.64 ± 45.52	486.47 ± 0.23	118.45 ± 40.21	103.24 ± 0.04
cluster_agg	4737.77 ± 44.13	78.66 ± 0.08	31730.67 ± 20.77	51.07 ± 0.04
single_agg	61.59 ± 38.97	1152.06 ± 0.41	285.74 ± 28.71	233.2 ± 0.07
prefix_agg	3125.93 ± 32.56	**19.61 ± 23.05**	14767.12 ± 21.75	**31.18 ± 39.18**
prefix_laststate	873.8 ± 43.06	56.39 ± 48.96	2801.64 ± 43.74	66.13 ± 23.5
single_laststate	46.86 ± 40.98	477.89 ± 0.14	358.81 ± 20.89	183.72 ± 0.04
prefix_index	117.24 ± 35.72	21.54 ± 49.8	3705.55 ± 46.9	43.97 ± 24.69
state_agg	116.0 ± 32.12	786.08 ± 0.22	62.24 ± 38.17	180.59 ± 0.04
state_laststate	**42.37 ± 35.7**	287.69 ± 0.07	**42.89 ± 27.98**	280.52 ± 0.06

Table B.8: Execution time for **SVM**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	150.92 ± 38.62	49.36 ± 0.03	580.01 ± 48.2	18.68 ± 0.01	130.53 ± 33.2	54.44 ± 0.03
cluster_agg	4534.44 ± 47.66	**30.24 ± 0.03**	62616.45 ± 21.58	**10.08 ± 0.01**	4470.5 ± 36.94	40.84 ± 0.05
single_agg	800.27 ± 23.19	158.93 ± 0.06	144.06 ± 25.9	23.72 ± 0.01	19.43 ± 42.68	172.13 ± 0.06
prefix_agg	8728.01 ± 21.34	39.13 ± 24.61	4968.95 ± 41.48	25.28 ± 30.95	6619.28 ± 25.3	**25.08 ± 32.06**
prefix_laststate	1339.36 ± 27.34	65.56 ± 20.71	950.35 ± 29.56	38.46 ± 21.17	999.92 ± 36.34	30.9 ± 25.15
single_laststate	112.29 ± 26.72	277.02 ± 0.08	1642.19 ± 40.33	31.26 ± 0.01	660.17 ± 40.48	351.17 ± 0.1
prefix_index	332.55 ± 47.51	40.31 ± 23.66	538.74 ± 23.93	24.14 ± 41.55	209.5 ± 31.27	44.05 ± 43.7
state_agg	**102.08 ± 45.47**	65.85 ± 0.02	400.42 ± 47.55	32.4 ± 0.01	**102.89 ± 42.08**	70.92 ± 0.02
state_laststate	120.63 ± 37.86	241.67 ± 0.07	**103.97 ± 27.46**	32.87 ± 0.01	219.13 ± 40.71	211.29 ± 0.06

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	237.84 ± 24.93	21.51 ± 0.01	2883.25 ± 41.34	47.43 ± 0.02	160.61 ± 27.67	329.69 ± 0.18
cluster_agg	74613.97 ± 34.17	**10.65 ± 0.01**	72707.65 ± 38.01	28.32 ± 0.02	11000.82 ± 27.8	45.02 ± 0.05
single_agg	202.68 ± 26.89	23.87 ± 0.01	279.28 ± 45.28	78.53 ± 0.02	794.84 ± 39.8	922.33 ± 0.35
prefix_agg	4953.66 ± 34.22	42.08 ± 48.55	14692.95 ± 48.9	**27.82 ± 23.76**	5153.2 ± 27.57	**26.62 ± 31.38**
prefix_laststate	953.74 ± 41.02	50.7 ± 21.35	2781.23 ± 40.91	56.91 ± 40.12	973.37 ± 48.09	45.17 ± 36.14
single_laststate	340.49 ± 30.86	29.04 ± 0.01	24916.22 ± 35.8	103.53 ± 0.02	451.26 ± 33.2	306.55 ± 0.09
prefix_index	545.74 ± 45.32	34.07 ± 44.35	3642.42 ± 49.46	35.6 ± 38.1	123.49 ± 31.76	45.63 ± 25.84
state_agg	**194.43 ± 27.62**	32.53 ± 0.01	**320.78 ± 24.31**	89.48 ± 0.02	12.99 ± 21.49	643.67 ± 0.21
state_laststate	523.35 ± 31.55	31.98 ± 0.01	324.83 ± 38.99	95.49 ± 0.02	**5.75 ± 47.08**	175.6 ± 0.05

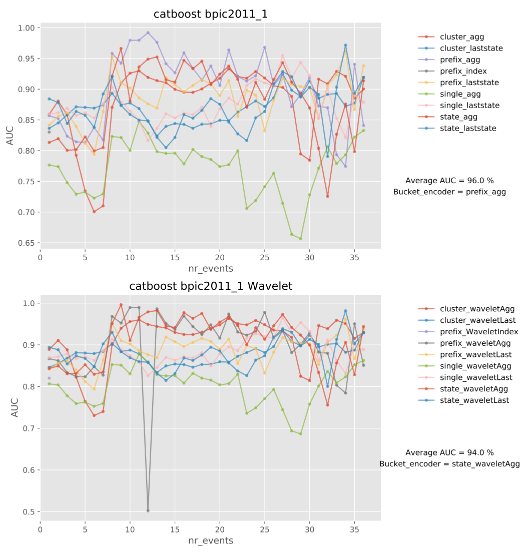
	traffic		**bpic2012_A**		**bpic2015_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	2626.8 ± 25.22	126.59 ± 0.02	106.17 ± 32.74	21.34 ± 0.01	320.85 ± 49.85	63.61 ± 0.03
cluster_agg	479674.83 ± 25.26	67.17 ± 0.03	46686.4 ± 40.81	**11.51 ± 0.01**	14603.4 ± 35.28	35.94 ± 0.04
single_agg	4390.07 ± 45.26	172.63 ± 0.02	382.32 ± 45.23	25.77 ± 0.01	805.75 ± 48.74	185.87 ± 0.07
prefix_agg	15231.69 ± 28.36	39.34 ± 27.9	5042.86 ± 27.2	24.62 ± 24.37	13997.3 ± 31.83	**32.37 ± 49.99**
prefix_laststate	2909.98 ± 23.39	58.57 ± 35.68	939.93 ± 21.61	35.55 ± 42.31	1900.29 ± 49.53	32.63 ± 46.2
single_laststate	3191.65 ± 46.45	213.8 ± 0.02	543.94 ± 28.11	30.47 ± 0.01	450.84 ± 34.8	464.0 ± 0.14
prefix_index	14258.92 ± 46.97	**38.67 ± 43.2**	538.03 ± 40.0	21.93 ± 33.88	452.8 ± 44.19	44.59 ± 26.31
state_agg	**1230.46 ± 45.34**	233.38 ± 0.03	**103.7 ± 46.3**	33.25 ± 0.01	**301.09 ± 44.67**	78.39 ± 0.02
state_laststate	2180.85 ± 23.85	253.46 ± 0.03	206.54 ± 39.41	32.79 ± 0.01	414.44 ± 20.93	263.44 ± 0.08

	bpic2011_3		**bpic2011_2**		**sepsis_2**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	28.31 ± 44.67	200.99 ± 0.09	137.12 ± 41.3	247.4 ± 0.13	2.96 ± 46.71	113.37 ± 0.04
cluster_agg	3426.69 ± 27.82	52.82 ± 0.05	18174.33 ± 28.81	48.1 ± 0.05	677.56 ± 36.58	45.07 ± 0.03
single_agg	267.47 ± 46.74	1120.66 ± 0.38	130.13 ± 43.03	461.25 ± 0.26	3.38 ± 41.97	300.99 ± 0.08
prefix_agg	2075.37 ± 34.84	**21.42 ± 27.0**	4281.8 ± 37.92	**28.44 ± 25.7**	472.04 ± 38.32	36.37 ± 43.36
prefix_laststate	521.65 ± 32.63	62.44 ± 21.03	1124.03 ± 21.39	45.7 ± 33.07	89.6 ± 24.35	**31.23 ± 42.92**
single_laststate	91.4 ± 28.55	417.95 ± 0.11	613.92 ± 42.57	328.21 ± 0.11	8.83 ± 37.39	450.69 ± 0.09
prefix_index	117.55 ± 35.42	48.27 ± 28.63	131.68 ± 47.74	43.1 ± 44.24	102.56 ± 29.41	46.32 ± 41.16
state_agg	8.91 ± 20.79	838.78 ± 0.25	18.32 ± 21.8	815.83 ± 0.26	**0.66 ± 22.2**	229.8 ± 0.05
state_laststate	**3.83 ± 37.41**	206.18 ± 0.05	**6.76 ± 37.48**	211.04 ± 0.06	2.87 ± 25.05	576.34 ± 0.1

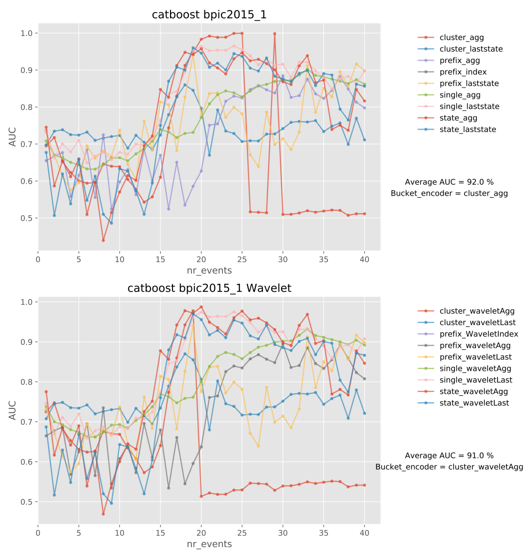
	bpic2015_1		**bpic2015_5**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	414.48 ± 25.63	71.75 ± 0.04	310.13 ± 24.67	75.35 ± 0.05	6.5 ± 31.58	95.33 ± 0.03
cluster_agg	4002.98 ± 45.04	45.74 ± 0.05	7358.17 ± 48.71	33.82 ± 0.04	1557.87 ± 25.94	38.05 ± 0.03
single_agg	277.52 ± 31.49	197.44 ± 0.07	757.68 ± 46.56	204.25 ± 0.08	13.35 ± 33.94	183.64 ± 0.06
prefix_agg	7629.7 ± 44.55	**28.17 ± 41.92**	13067.22 ± 45.09	**31.37 ± 29.05**	1422.54 ± 30.58	39.33 ± 23.63
prefix_laststate	1098.43 ± 41.98	55.52 ± 28.6	1717.45 ± 42.32	34.93 ± 28.18	266.47 ± 33.27	43.56 ± 42.96
single_laststate	395.73 ± 48.05	328.97 ± 0.1	352.38 ± 34.29	385.17 ± 0.12	12.46 ± 24.54	349.98 ± 0.08
prefix_index	314.17 ± 35.85	42.84 ± 33.74	468.84 ± 42.43	40.49 ± 35.51	106.78 ± 30.97	**27.76 ± 23.84**
state_agg	**201.25 ± 43.37**	81.35 ± 0.02	**97.76 ± 42.73**	82.27 ± 0.02	**1.06 ± 21.94**	184.01 ± 0.04
state_laststate	314.03 ± 29.52	311.16 ± 0.09	100.81 ± 37.24	227.92 ± 0.07	4.98 ± 33.63	507.81 ± 0.1

	bpic2017_C		**production**		**sepsis_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	1096.63 ± 48.9	41.75 ± 0.02	3.54 ± 44.8	116.44 ± 0.03	7.57 ± 28.37	91.69 ± 0.03
cluster_agg	674260.87 ± 41.68	**24.32 ± 0.02**	1440.53 ± 30.46	38.51 ± 0.03	944.89 ± 22.52	38.29 ± 0.03
single_agg	1759.09 ± 26.8	59.73 ± 0.02	2.98 ± 24.36	110.25 ± 0.02	11.25 ± 32.8	221.07 ± 0.07
prefix_agg	14694.85 ± 38.95	35.84 ± 40.91	518.19 ± 29.52	31.78 ± 39.41	130.24 ± 45.19	**18.71 ± 24.82**
prefix_laststate	2773.99 ± 43.09	53.57 ± 21.03	191.03 ± 48.51	59.91 ± 46.1	258.48 ± 43.79	52.05 ± 30.34
single_laststate	9308.12 ± 28.34	74.23 ± 0.02	3.43 ± 36.51	125.63 ± 0.02	102.3 ± 49.87	527.69 ± 0.15
prefix_index	3675.99 ± 21.26	39.06 ± 36.6	23.54 ± 40.55	**22.44 ± 27.62**	103.34 ± 39.41	47.2 ± 42.45
state_agg	**1052.67 ± 25.73**	69.94 ± 0.02	**0.68 ± 44.49**	267.18 ± 0.05	**0.96 ± 43.89**	179.84 ± 0.04
state_laststate	1202.9 ± 23.72	73.47 ± 0.01	1.08 ± 29.11	245.56 ± 0.04	2.67 ± 40.99	263.81 ± 0.06

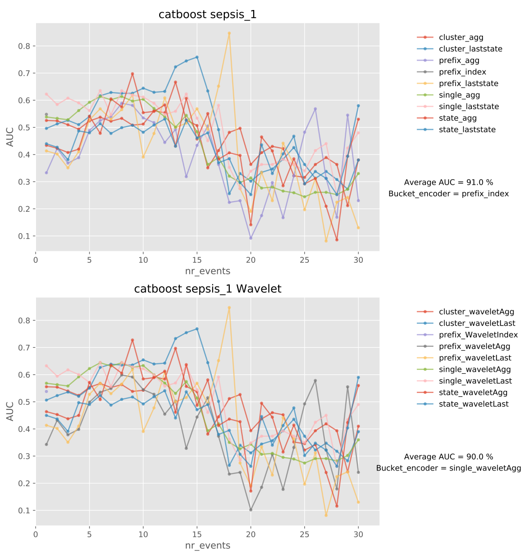
	bpic2011_1		**bpic2017_R**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	79.51 ± 41.74	342.11 ± 0.18	861.37 ± 23.84	44.96 ± 0.02
cluster_agg	7855.79 ± 31.64	52.32 ± 0.05	274901.95 ± 47.93	28.07 ± 0.02
single_agg	88.07 ± 27.75	215.46 ± 0.08	346.81 ± 34.68	77.54 ± 0.02
prefix_agg	3777.45 ± 48.79	**22.44 ± 36.91**	15944.05 ± 47.06	**25.1 ± 41.01**
prefix_laststate	918.55 ± 29.09	56.73 ± 35.96	5274.55 ± 43.14	65.71 ± 27.4
single_laststate	204.35 ± 41.38	230.37 ± 0.06	171.95 ± 40.84	74.76 ± 0.02
prefix_index	118.69 ± 37.79	27.94 ± 31.22	3727.86 ± 44.23	28.31 ± 28.11
state_agg	11.62 ± 29.59	809.68 ± 0.24	476.89 ± 20.52	80.22 ± 0.02
state_laststate	**6.25 ± 31.19**	268.88 ± 0.07	**83.18 ± 42.07**	73.63 ± 0.01



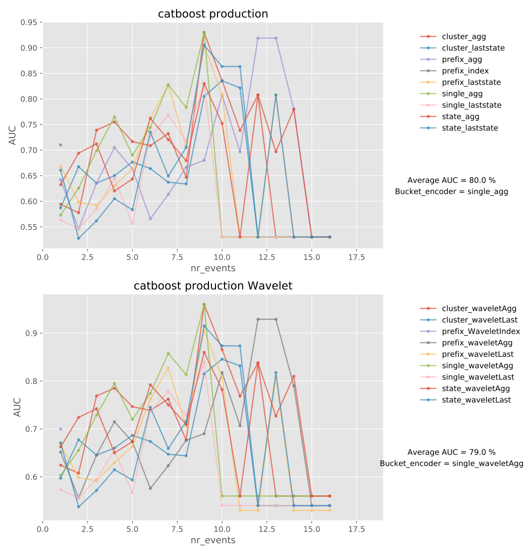
(a) bpic2011_1



(b) bpic2015_1

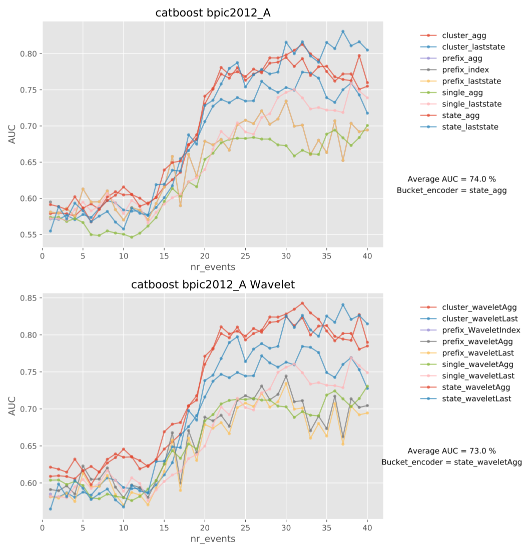


(c) sepsis_1

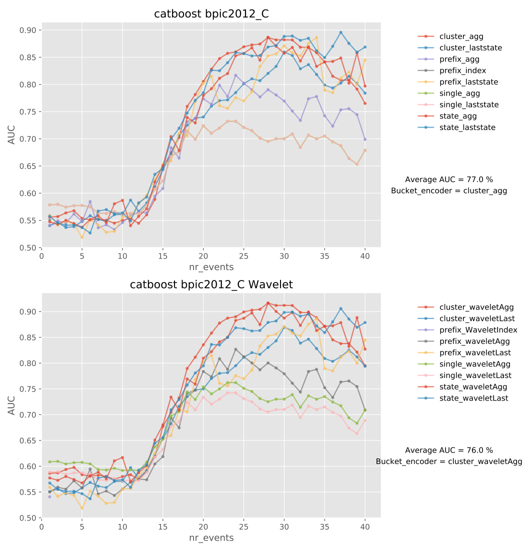


(d) production

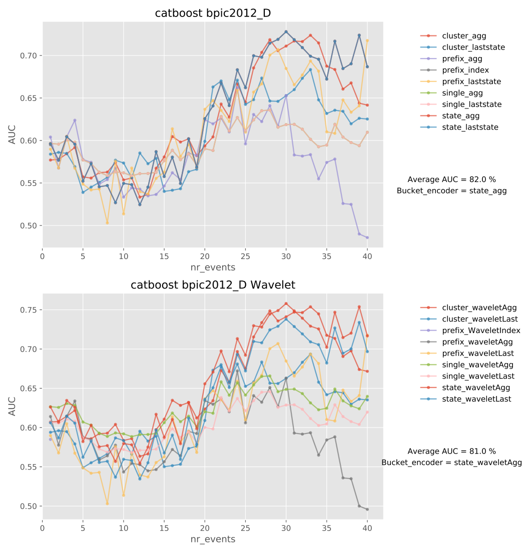
Figure B.10: Comparison of CatBoost before and after adding Wavelet encoding



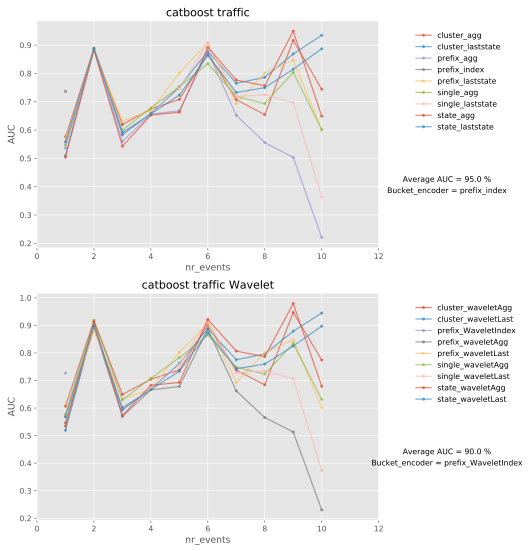
(a) bpic2012_A



(b) bpic2012_C

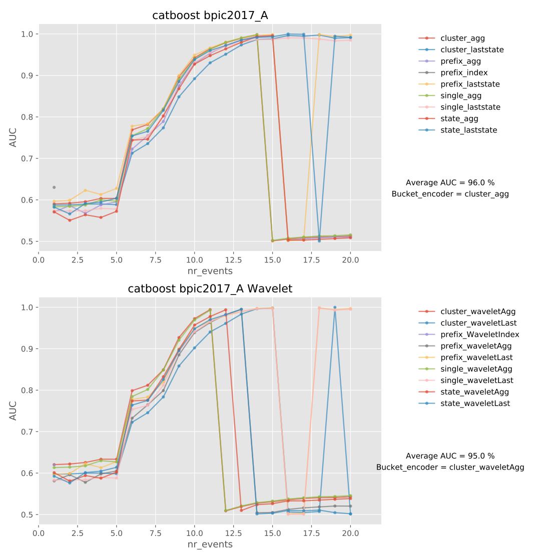


(c) bpic2012_D

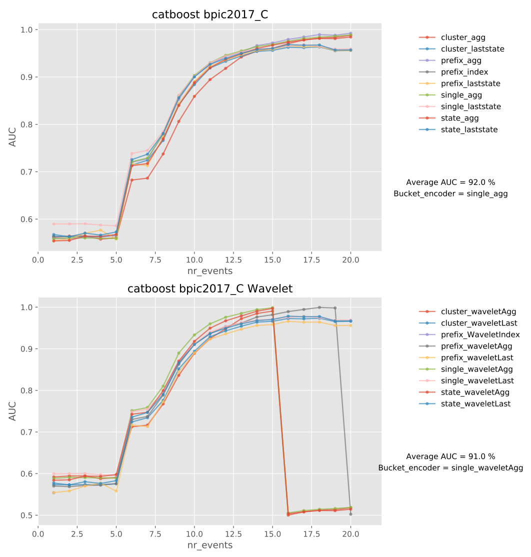


(d) traffic

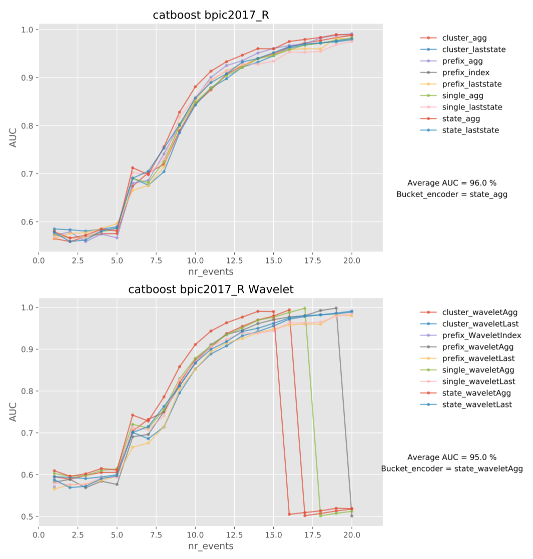
Figure B.11: Comparison of CatBoost before and after adding Wavelet encoding *(continued)*



(a) bpic2017_A

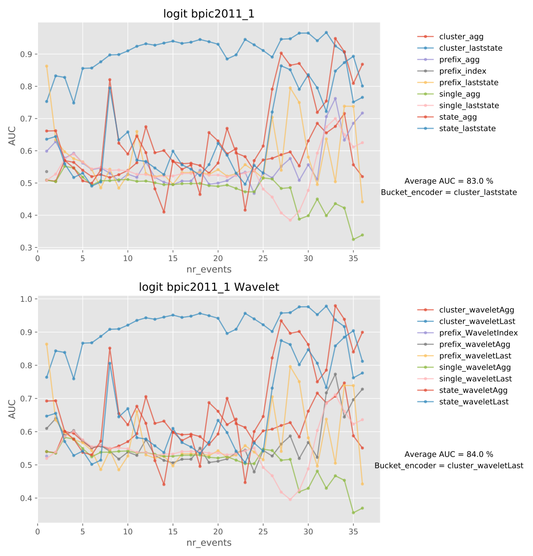


(b) bpic2017_C

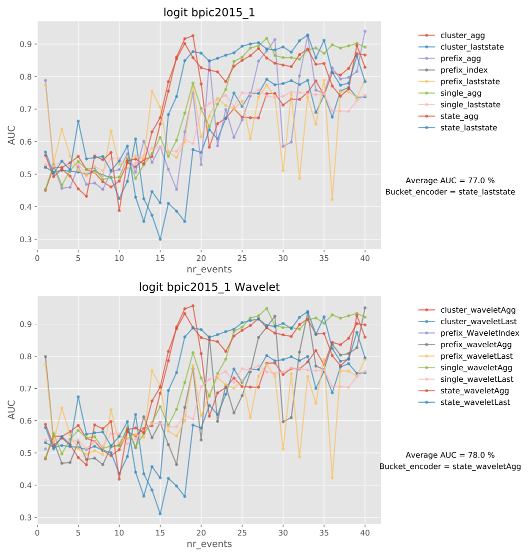


(c) bpic2017_R

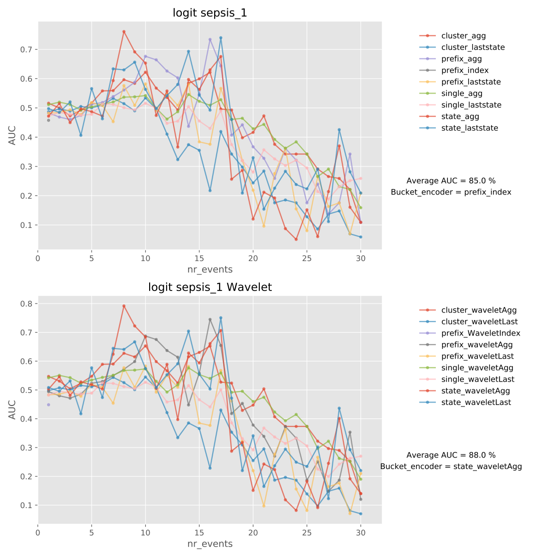
Figure B.12: Comparison of CatBoost before and after adding Wavelet encoding *(continued)*



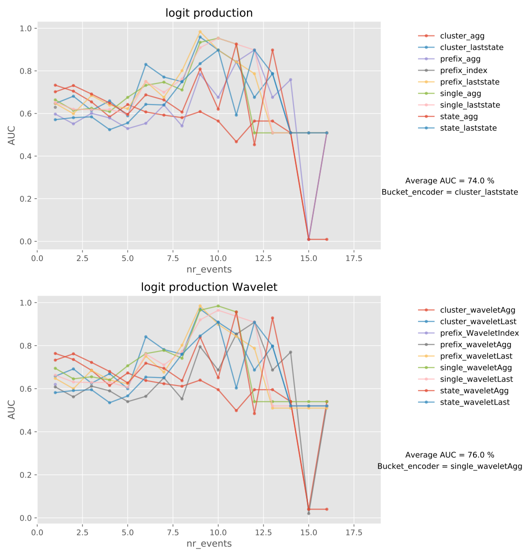
(a) bpic2011_1



(b) bpic2015_1

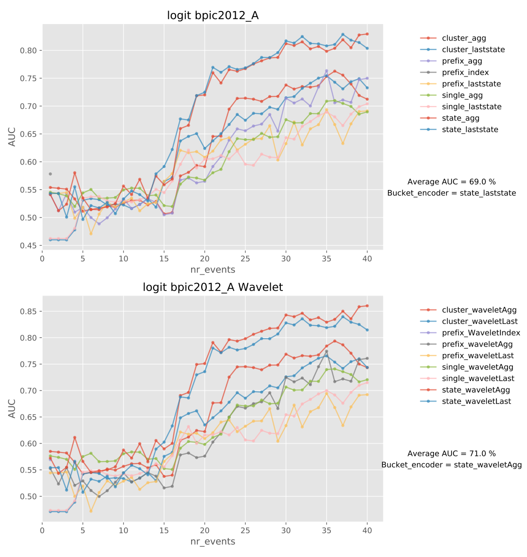


(c) sepsis_1

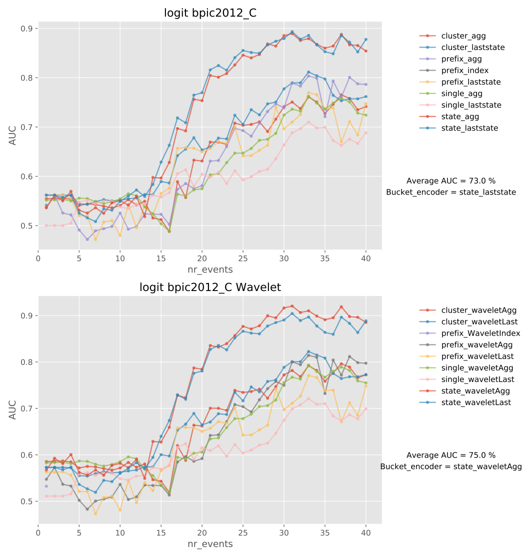


(d) production

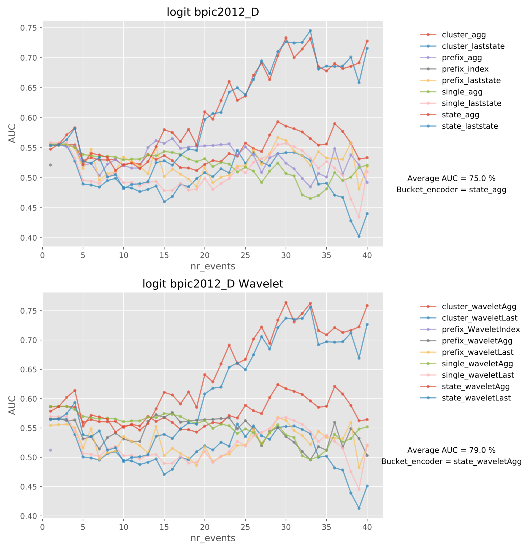
Figure B.13: Comparison of LR before and after adding Wavelet encoding



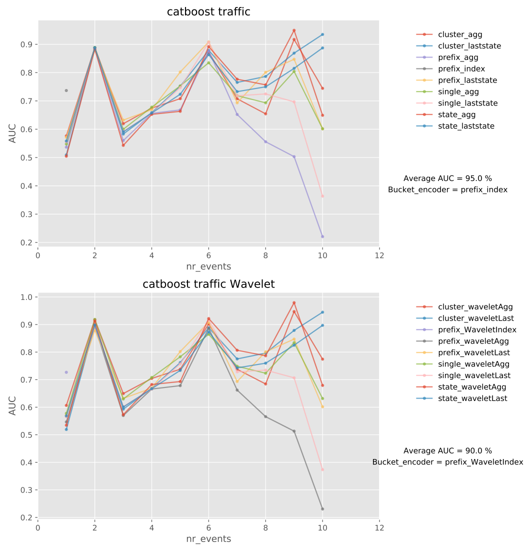
(a) bpic2012_A



(b) bpic2012_C

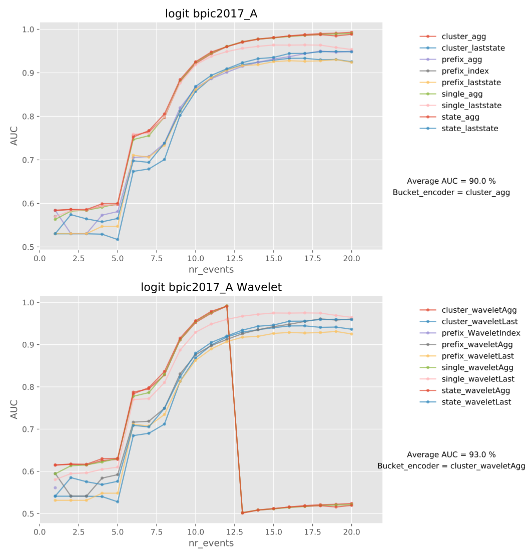


(c) bpic2012_D

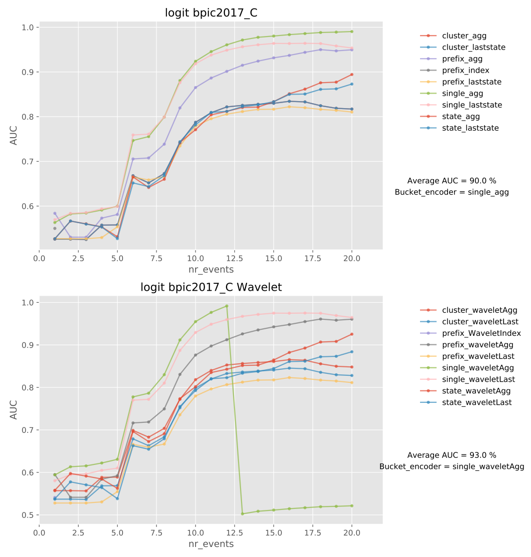


(d) traffic

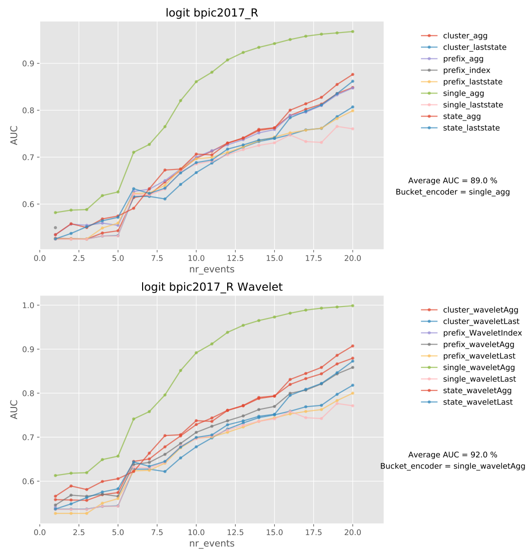
Figure B.14: Comparison of LR before and after adding Wavelet encoding *(continued)*



(a) bpic2017_A

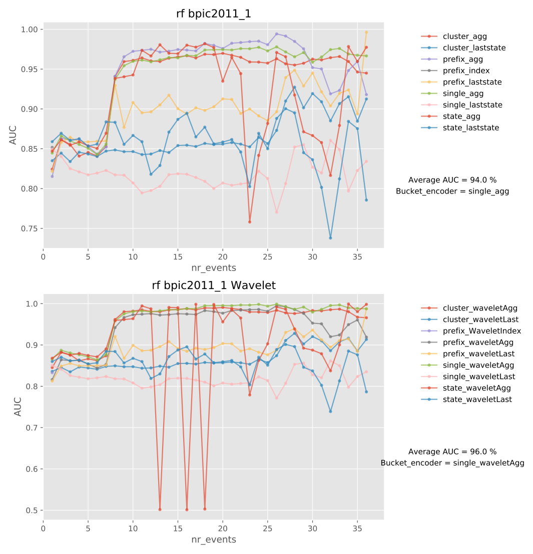


(b) bpic2017_C

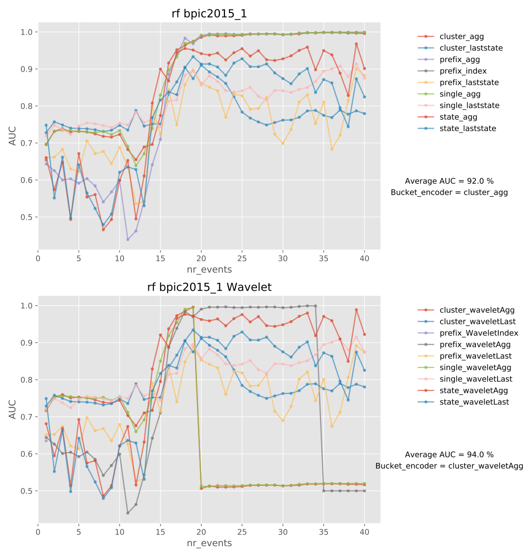


(c) bpic2017_R

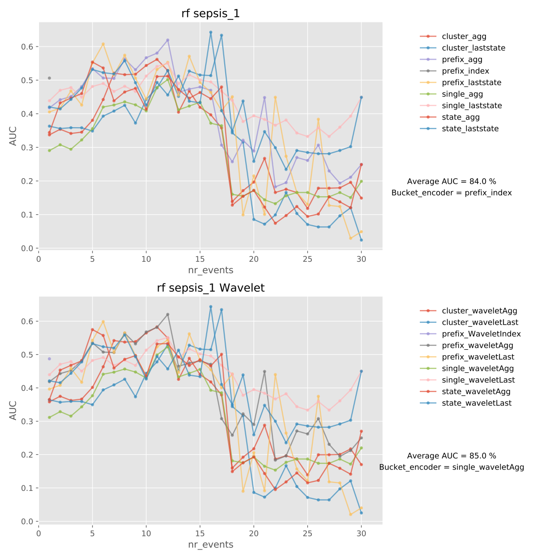
Figure B.15: Comparison of LR before and after adding Wavelet encoding *(continued)*



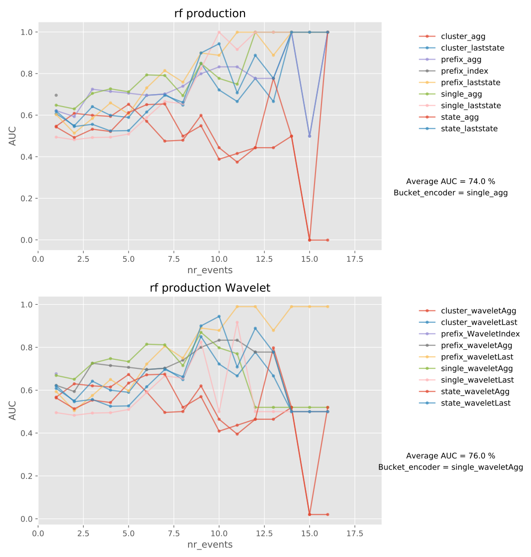
(a) bpic2011_1



(b) bpic2015_1

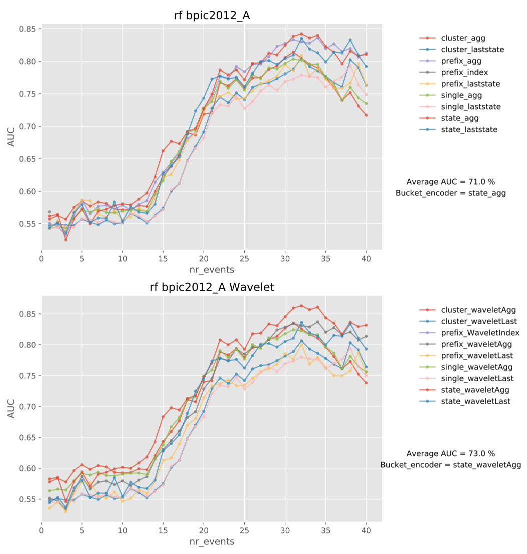


(c) sepsis_1

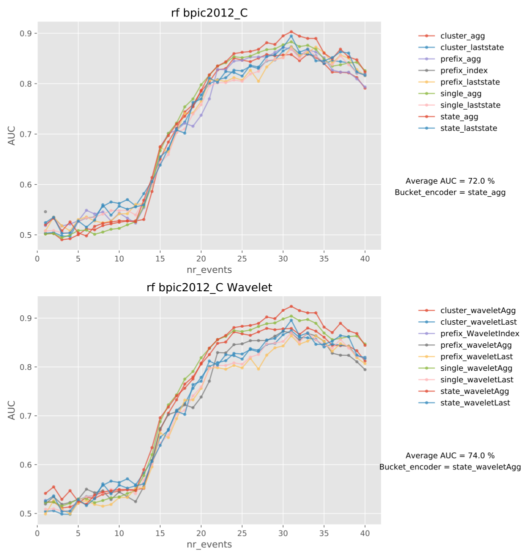


(d) production

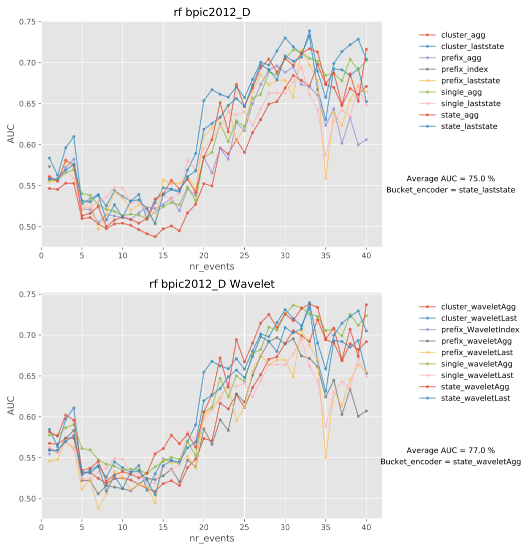
Figure B.16: Comparison of RF before and after adding Wavelet encoding



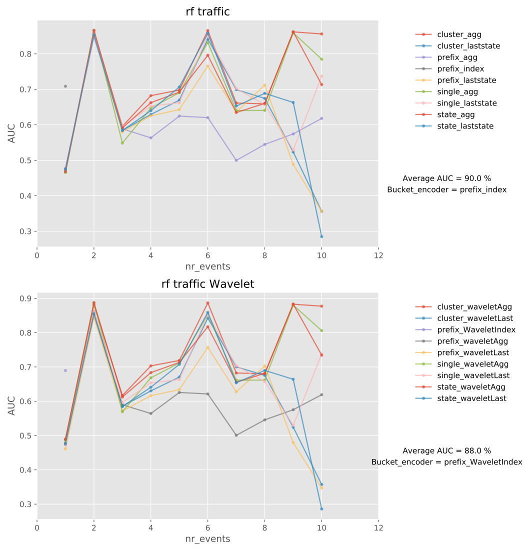
(a) bpic2012_A



(b) bpic2012_C

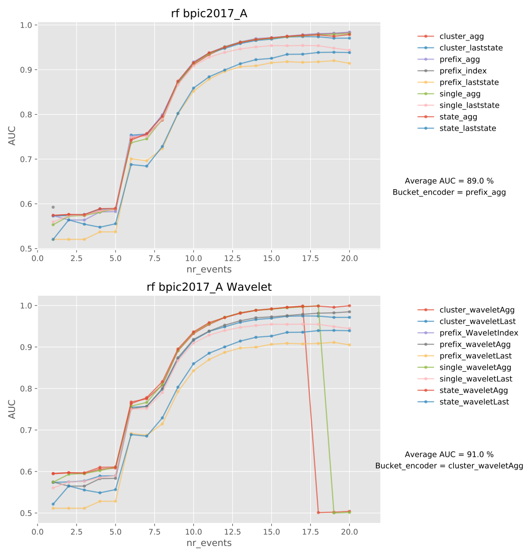


(c) bpic2012_D

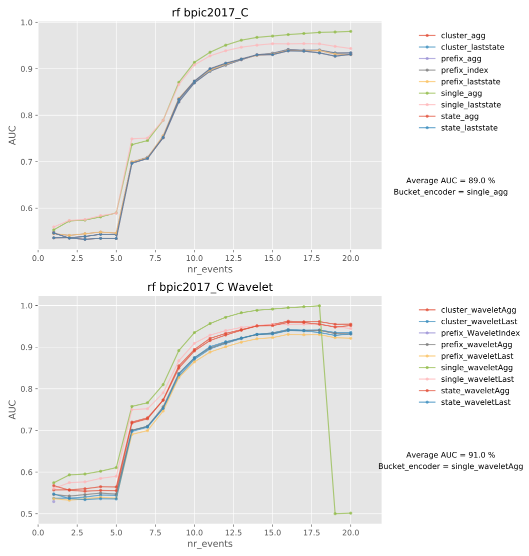


(d) traffic

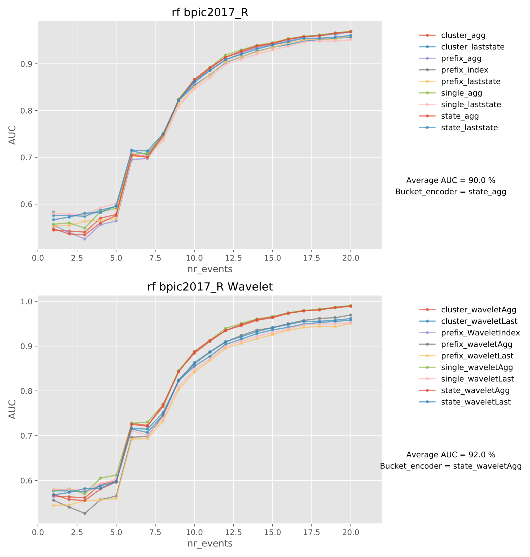
Figure B.17: Comparison of RF before and after adding Wavelet encoding *(continued)*



(a) bpic2017_A

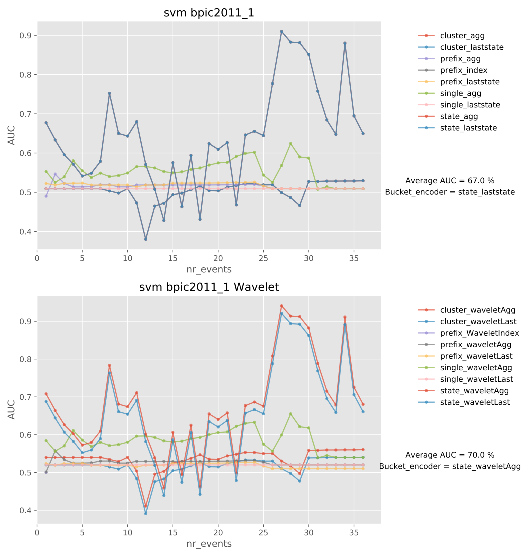


(b) bpic2017_C

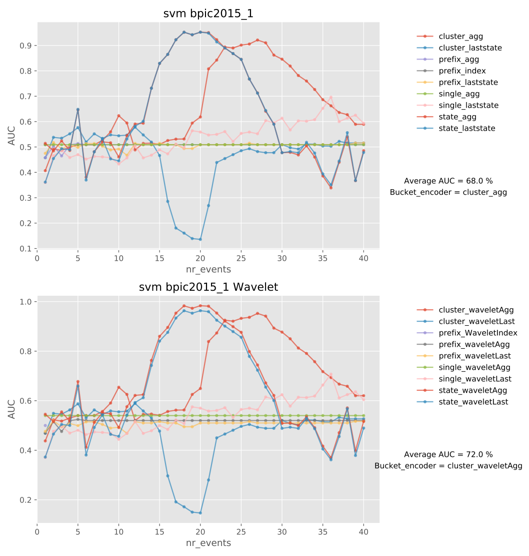


(c) bpic2017_R

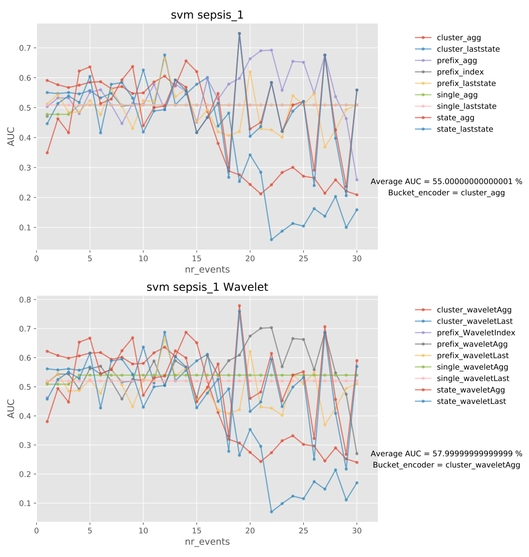
Figure B.18: Comparison of RF before and after adding Wavelet encoding *(continued)*



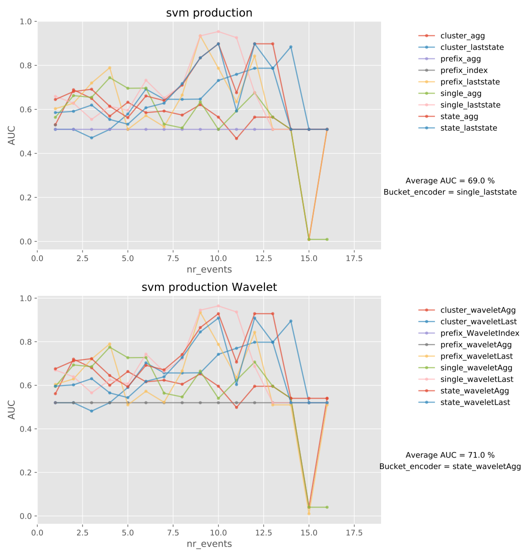
(a) bpic2011_1



(b) bpic2015_1

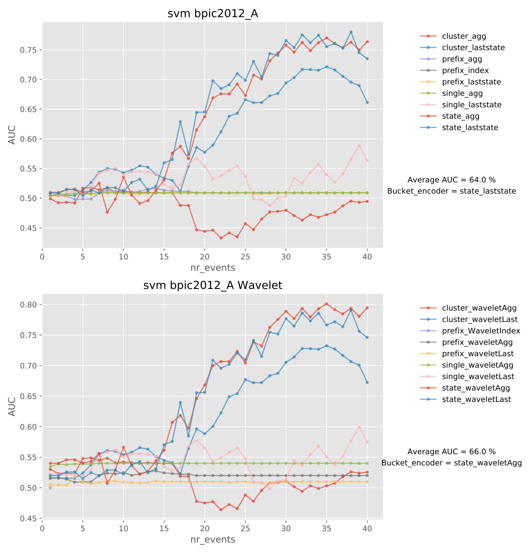


(c) sepsis_1

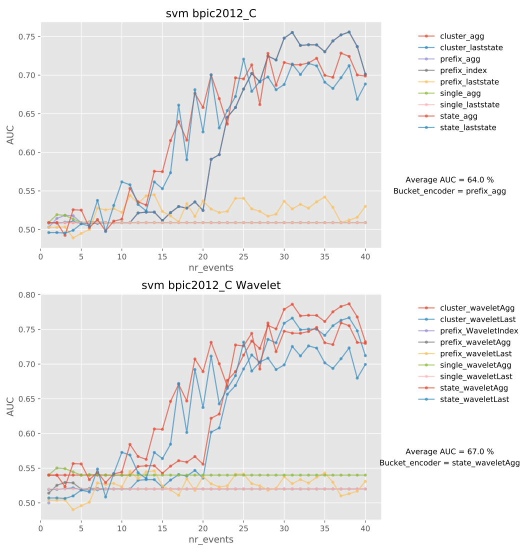


(d) production

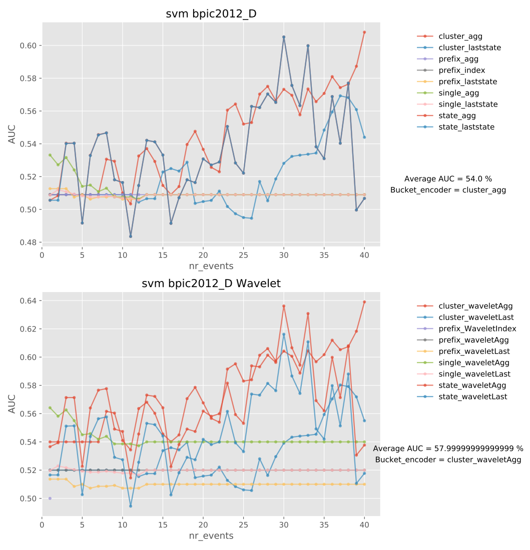
Figure B.19: Comparison of SVM before and after adding Wavelet encoding



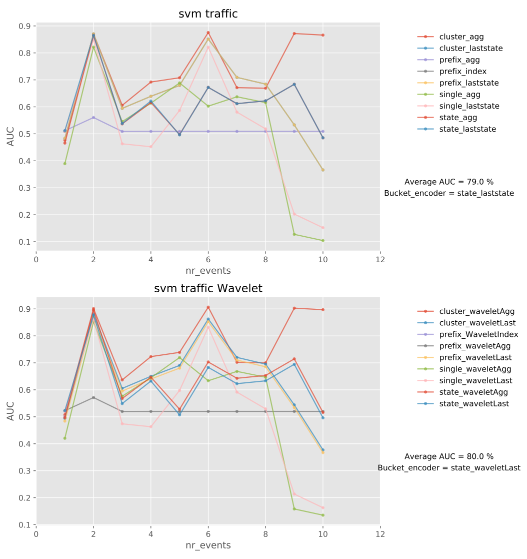
(a) bpic2012_A



(b) bpic2012_C

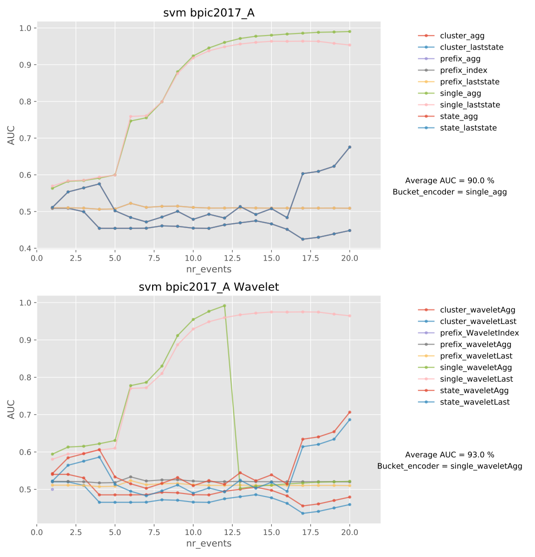


(c) bpic2012_D

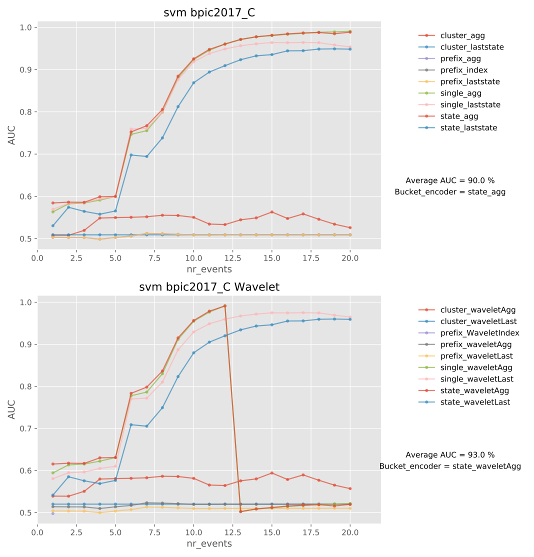


(d) traffic

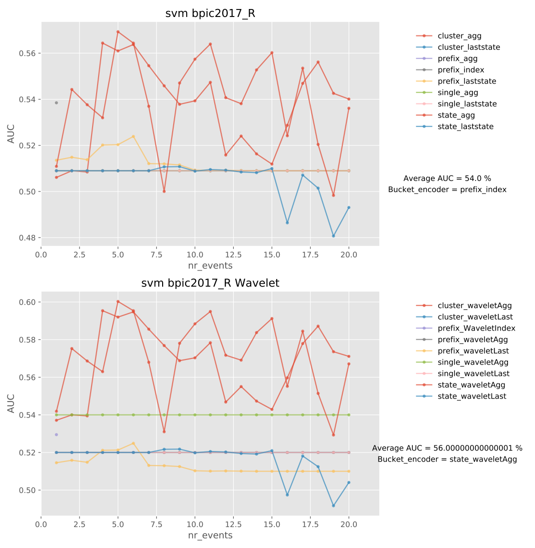
Figure B.20: Comparison of SVM before and after adding Wavelet encoding *(continued)*



(a) bpic2017_A



(b) bpic2017_C



(c) bpic2017_R

Figure B.21: Comparison of SVM before and after adding Wavelet encoding *(continued)*

Table B.9: Overall AUC (F-score) for **Wavelet plus XGBoost**

	xgboost					
	bpic2015_2	bpic2015_4	bpic2012_c	bpic2011_4	bpic2012_d	traffic
cluster_waveletAgg	0.84 (0.64)	0.71 (0.3)	0.74 (0.48)	0.71 (0.58)	0.64 (0.21)	0.7 (0.7)
prefix_WaveletIndex	0.74 (0.39)	0.67 (0.25)	0.56 (0.34)	0.79 (0.6)	0.59 (0.13)	0.69 (0.19)
prefix_waveletLast	0.83 (0.43)	0.68 (0.24)	0.68 (0.44)	0.81 (0.66)	0.58 (0.19)	0.68 (0.69)
state_waveletAgg	0.91 (0.4)	0.67 (0.22)	0.74 (0.51)	0.8 (0.59)	**0.65 (0.25)**	0.69 (0.69)
single_waveletLast	0.86 (0.48)	0.73 (0.25)	0.72 (0.47)	0.81 (0.64)	0.63 (0.19)	0.68 (0.68)
single_waveletAgg	0.56 (0.53)	0.65 (0.23)	**0.75 (0.51)**	0.88 (0.69)	0.62 (0.22)	0.7 (0.72)
state_waveletLast	0.88 (0.35)	0.7 (0.28)	0.71 (0.47)	0.76 (0.66)	0.63 (0.26)	0.68 (0.68)
prefix_waveletAgg	0.54 (0.51)	0.67 (0.23)	0.69 (0.45)	0.85 (0.67)	0.6 (0.21)	0.67 (0.64)
cluster_waveletLast	0.91 (0.46)	0.7 (0.24)	0.71 (0.46)	0.78 (0.6)	0.6 (0.21)	0.7 (0.7)

	xgboost					
	bpic2012_a	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1
cluster_waveletAgg	0.72 (0.67)	0.65 (0.4)	0.92 (0.69)	0.55 (0.52)	0.56 (0.06)	0.62 (0.63)
prefix_WaveletIndex	0.6 (0.49)	0.63 (0.33)	0.85 (0.68)	0.81 (0.85)	**0.88 (0.55)**	0.54 (0.27)
prefix_waveletLast	0.65 (0.63)	0.63 (0.27)	0.87 (0.72)	0.84 (0.82)	0.53 (0.03)	0.74 (0.39)
state_waveletAgg	**0.73 (0.69)**	0.66 (0.32)	0.86 (0.64)	0.67 (0.79)	0.39 (0.06)	0.8 (0.56)
single_waveletLast	0.66 (0.66)	0.6 (0.34)	0.87 (0.68)	0.86 (0.86)	0.54 (0.04)	0.73 (0.43)
single_waveletAgg	0.72 (0.64)	0.67 (0.34)	0.91 (0.76)	0.55 (0.99)	0.56 (0.06)	0.63 (0.59)
state_waveletLast	0.7 (0.66)	0.61 (0.29)	0.84 (0.59)	0.55 (0.77)	0.37 (0.04)	0.79 (0.51)
prefix_waveletAgg	0.68 (0.67)	0.65 (0.31)	0.92 (0.7)	0.5 (0.89)	0.54 (0.04)	0.63 (0.53)
cluster_waveletLast	0.67 (0.65)	0.6 (0.22)	0.91 (0.76)	0.86 (0.85)	0.54 (0.04)	0.74 (0.39)

	xgboost					
	bpic2015_5	sepsis_3	bpic2017_c	production	bpic2017_r	sepsis_1
cluster_waveletAgg	0.68 (0.44)	0.68 (0.12)	**0.84 (0.77)**	**0.74 (0.61)**	0.73 (0.51)	0.44 (0.1)
prefix_WaveletIndex	0.63 (0.46)	0.68 (0.17)	0.55 (0.46)	0.66 (0.66)	0.6 (0.05)	0.56 (0.02)
prefix_waveletLast	0.68 (0.45)	0.7 (0.3)	0.81 (0.74)	0.65 (0.53)	0.81 (0.48)	0.52 (0.1)
state_waveletAgg	0.69 (0.47)	0.71 (0.06)	0.72 (0.78)	**0.74 (0.65)**	0.79 (0.52)	0.53 (0.16)
single_waveletLast	0.71 (0.46)	0.72 (0.14)	0.81 (0.75)	0.62 (0.58)	0.82 (0.45)	0.43 (0.1)
single_waveletAgg	0.71 (0.42)	0.7 (0.27)	0.72 (0.77)	0.71 (0.63)	0.75 (0.45)	0.5 (0.15)
state_waveletLast	0.67 (0.47)	0.72 (0.28)	0.83 (0.77)	0.58 (0.55)	0.81 (0.49)	0.46 (0.12)
prefix_waveletAgg	0.7 (0.48)	0.7 (0.19)	0.82 (0.76)	**0.74 (0.6)**	0.81 (0.5)	0.47 (0.1)
cluster_waveletLast	0.69 (0.45)	0.74 (0.31)	0.82 (0.75)	0.63 (0.55)	0.81 (0.47)	0.45 (0.08)

	xgboost	
	bpic2011_1	bpic2017_a
cluster_waveletAgg	0.67 (0.89)	0.65 (0.64)
prefix_WaveletIndex	0.75 (0.68)	0.57 (0.44)
prefix_waveletLast	0.85 (0.75)	0.83 (0.73)
state_waveletAgg	0.82 (0.84)	0.66 (0.69)
single_waveletLast	0.85 (0.78)	0.69 (0.74)
single_waveletAgg	0.65 (0.88)	0.66 (0.68)
state_waveletLast	0.78 (0.7)	0.73 (0.75)
prefix_waveletAgg	0.88 (0.82)	0.66 (0.73)
cluster_waveletLast	0.87 (0.78)	0.69 (0.71)

Table B.10: Overall AUC (F-score) for **Wavelet plus Logit**

	logit					
	bpic2015_2	bpic2015_4	bpic2012_c	bpic2011_4	bpic2012_d	traffic
cluster_waveletAgg	0.91 (0.12)	0.68 (0.04)	0.65 (0.14)	0.87 (0.76)	0.57 (0.04)	0.69 (0.66)
prefix_WaveletIndex	0.68 (0.25)	0.68 (0.0)	0.53 (0.0)	0.65 (0.12)	0.51 (0.0)	0.57 (0.0)
prefix_waveletLast	0.75 (0.01)	0.62 (0.01)	0.61 (0.15)	0.91 (0.79)	0.53 (0.01)	0.61 (0.55)
state_waveletAgg	0.93 (0.52)	0.68 (0.04)	0.72 (0.3)	0.68 (0.49)	0.63 (0.13)	0.68 (0.64)
single_waveletLast	0.76 (0.02)	0.75 (0.02)	0.6 (0.09)	0.64 (0.02)	0.52 (0.02)	0.61 (0.55)
single_waveletAgg	0.92 (0.04)	0.66 (0.04)	0.63 (0.13)	0.82 (0.69)	0.56 (0.04)	0.63 (0.54)
state_waveletLast	0.88 (0.38)	0.68 (0.02)	0.71 (0.29)	0.69 (0.5)	0.6 (0.11)	0.64 (0.63)
prefix_waveletAgg	0.9 (0.11)	0.64 (0.02)	0.61 (0.18)	0.61 (0.06)	0.55 (0.02)	0.65 (0.54)
cluster_waveletLast	0.68 (0.34)	**0.77** (**0.12**)	0.65 (0.18)	0.89 (0.76)	0.51 (0.02)	0.64 (0.62)

	logit					
	bpic2012_a	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1
cluster_waveletAgg	0.64 (0.41)	0.49 (0.04)	0.77 (0.63)	0.65 (0.81)	0.54 (0.04)	0.69 (0.13)
prefix_WaveletIndex	0.57 (0.09)	0.56 (0.06)	0.92 (0.8)	0.93 (0.9)	0.87 (0.56)	0.51 (0.06)
prefix_waveletLast	0.59 (0.41)	0.48 (0.01)	0.77 (0.06)	0.54 (0.78)	0.51 (0.01)	0.64 (0.24)
state_waveletAgg	0.69 (0.49)	0.65 (0.12)	0.88 (0.77)	0.55 (0.75)	0.37 (0.04)	0.74 (0.45)
single_waveletLast	0.59 (0.57)	0.47 (0.02)	0.95 (0.79)	0.63 (0.79)	0.52 (0.02)	0.65 (0.18)
single_waveletAgg	0.62 (0.47)	0.64 (0.14)	0.94 (0.83)	0.61 (0.81)	0.54 (0.04)	0.72 (0.07)
state_waveletLast	0.67 (0.45)	0.65 (0.1)	0.77 (0.58)	0.52 (0.69)	0.35 (0.02)	0.72 (0.46)
prefix_waveletAgg	0.6 (0.44)	0.62 (0.12)	0.93 (0.8)	0.59 (0.79)	0.52 (0.02)	0.65 (0.24)
cluster_waveletLast	0.61 (0.38)	0.64 (0.12)	0.86 (0.02)	0.94 (0.88)	0.52 (0.02)	0.6 (0.31)

	logit					
	bpic2015_5	sepsis_3	bpic2017_c	production	bpic2017_r	sepsis_1
cluster_waveletAgg	0.69 (0.33)	0.61 (0.04)	0.75 (0.67)	0.66 (0.56)	0.72 (0.04)	0.56 (0.04)
prefix_WaveletIndex	0.66 (0.31)	0.59 (0.0)	0.54 (0.39)	0.62 (0.55)	0.54 (0.01)	0.45 (0.0)
prefix_waveletLast	0.65 (0.23)	0.6 (0.01)	0.72 (0.61)	0.68 (0.54)	0.67 (0.01)	0.48 (0.06)
state_waveletAgg	0.66 (0.32)	0.54 (0.04)	0.76 (0.67)	0.7 (0.61)	0.73 (0.07)	0.54 (0.07)
single_waveletLast	0.68 (0.38)	0.73 (0.3)	**0.84** (**0.68**)	0.69 (0.55)	0.67 (0.02)	0.48 (0.03)
single_waveletAgg	0.67 (0.28)	**0.77** (**0.25**)	0.67 (0.71)	0.71 (0.57)	**0.84** (**0.4**)	0.53 (0.08)
state_waveletLast	0.69 (0.35)	0.71 (0.09)	0.74 (0.63)	0.63 (0.56)	0.69 (0.04)	0.53 (0.03)
prefix_waveletAgg	0.67 (0.33)	0.68 (0.35)	0.81 (0.58)	0.61 (0.56)	0.7 (0.02)	0.53 (0.13)
cluster_waveletLast	0.69 (0.38)	0.64 (0.02)	0.73 (0.65)	0.71 (0.63)	0.67 (0.02)	0.45 (0.02)

	logit	
	bpic2011_1	bpic2017_a
cluster_waveletAgg	0.6 (0.55)	0.67 (0.7)
prefix_WaveletIndex	0.53 (0.68)	0.56 (0.01)
prefix_waveletLast	0.57 (0.61)	0.79 (0.56)
state_waveletAgg	0.65 (0.71)	0.68 (0.73)
single_waveletLast	0.54 (0.64)	0.84 (0.68)
single_waveletAgg	0.52 (0.61)	0.67 (0.71)
state_waveletLast	0.63 (0.69)	0.81 (0.58)
prefix_waveletAgg	0.56 (0.58)	0.81 (0.58)
cluster_waveletLast	0.91 (0.74)	0.79 (0.57)

Table B.11: Overall AUC (F-score) for **Wavelet plus RF**

	rf					
	bpic2015_2	bpic2015_4	bpic2012_c	bpic2011_4	bpic2012_d	traffic
cluster_waveletAgg	0.52 (0.97)	0.71 (0.17)	0.7 (0.43)	0.89 (0.76)	0.58 (0.23)	0.68 (0.68)
prefix_WaveletIndex	0.72 (0.47)	0.68 (0.24)	0.53 (0.22)	0.88 (0.73)	0.55 (0.04)	0.69 (0.09)
prefix_waveletLast	0.91 (0.23)	0.7 (0.23)	0.68 (0.39)	0.88 (0.75)	0.58 (0.16)	0.63 (0.65)
state_waveletAgg	0.91 (0.42)	0.69 (0.15)	0.72 (0.44)	0.85 (0.68)	0.61 (0.19)	0.68 (0.68)
single_waveletLast	0.92 (0.24)	0.69 (0.21)	0.68 (0.44)	0.89 (0.74)	0.58 (0.18)	0.66 (0.66)
single_waveletAgg	0.52 (1.0)	0.69 (0.22)	0.71 (0.42)	0.89 (0.77)	0.61 (0.23)	0.67 (0.68)
state_waveletLast	0.84 (0.34)	0.69 (0.24)	0.69 (0.42)	0.84 (0.68)	0.6 (0.18)	0.65 (0.66)
prefix_waveletAgg	0.98 (0.96)	0.66 (0.19)	0.69 (0.41)	0.88 (0.77)	0.58 (0.18)	0.63 (0.52)
cluster_waveletLast	0.7 (0.55)	0.67 (0.18)	0.69 (0.42)	0.86 (0.71)	0.61 (0.09)	0.65 (0.66)

	rf					
	bpic2012_a	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1
cluster_waveletAgg	0.69 (0.67)	0.65 (0.27)	0.93 (0.76)	0.51 (0.51)	0.52 (0.02)	0.65 (0.75)
prefix_WaveletIndex	0.55 (0.43)	0.59 (0.24)	0.93 (0.79)	0.9 (0.85)	0.83 (0.67)	0.64 (0.17)
prefix_waveletLast	0.65 (0.6)	0.61 (0.23)	0.94 (0.74)	0.92 (0.87)	0.49 (−0.01)	0.73 (0.4)
state_waveletAgg	0.7 (0.66)	0.66 (0.2)	0.91 (0.78)	0.74 (0.82)	0.35 (0.02)	0.82 (0.66)
single_waveletLast	0.65 (0.62)	0.62 (0.23)	0.95 (0.74)	0.9 (0.83)	0.5 (0.0)	0.81 (0.53)
single_waveletAgg	0.69 (0.66)	0.65 (0.21)	0.96 (0.79)	0.52 (0.89)	0.52 (0.02)	0.66 (0.69)
state_waveletLast	0.68 (0.62)	0.62 (0.28)	0.93 (0.81)	0.62 (0.75)	0.33 (0.0)	0.77 (0.49)
prefix_waveletAgg	0.68 (0.64)	0.63 (0.22)	0.9 (0.73)	0.5 (0.91)	0.5 (0.0)	0.76 (0.65)
cluster_waveletLast	0.65 (0.63)	0.62 (0.08)	0.93 (0.74)	0.91 (0.84)	0.5 (0.0)	0.78 (0.47)

	rf					
	bpic2015_5	sepsis_3	bpic2017_c	production	bpic2017_r	sepsis_1
cluster_waveletAgg	0.7 (0.52)	0.74 (0.31)	0.81 (0.74)	0.57 (0.5)	0.82 (0.49)	0.43 (0.02)
prefix_WaveletIndex	0.66 (0.44)	0.7 (0.33)	0.53 (0.45)	0.68 (0.56)	0.56 (−0.0)	0.49 (−0.02)
prefix_waveletLast	0.71 (0.33)	0.73 (0.2)	0.78 (0.71)	0.68 (0.54)	0.79 (0.45)	0.46 (0.02)
state_waveletAgg	0.69 (0.48)	0.72 (0.1)	0.81 (0.74)	0.56 (0.46)	0.82 (0.49)	0.45 (0.02)
single_waveletLast	0.67 (0.47)	0.74 (0.14)	0.82 (0.66)	0.55 (0.52)	0.8 (0.45)	0.47 (0.0)
single_waveletAgg	0.71 (0.43)	0.74 (0.37)	0.8 (0.69)	0.71 (0.63)	0.83 (0.45)	0.38 (0.02)
state_waveletLast	0.72 (0.22)	0.69 (0.3)	0.79 (0.72)	0.62 (0.55)	0.81 (0.46)	0.46 (0.0)
prefix_waveletAgg	0.68 (0.48)	0.72 (0.3)	0.79 (0.72)	0.69 (0.6)	0.8 (0.47)	0.48 (0.02)
cluster_waveletLast	0.69 (0.46)	0.72 (0.29)	0.79 (0.72)	0.63 (0.55)	0.8 (0.46)	0.39 (0.01)

	rf	
	bpic2011_1	bpic2017_a
cluster_waveletAgg	0.95 (0.86)	0.78 (0.7)
prefix_WaveletIndex	0.83 (0.71)	0.57 (0.37)
prefix_waveletLast	0.89 (0.74)	0.77 (0.54)
state_waveletAgg	0.89 (0.87)	**0.85** (**0.71**)
single_waveletLast	0.82 (0.71)	0.82 (0.66)
single_waveletAgg	**0.96** (**0.89**)	0.8 (0.69)
state_waveletLast	0.86 (0.72)	0.79 (0.56)
prefix_waveletAgg	0.94 (0.88)	0.83 (0.71)
cluster_waveletLast	0.86 (0.73)	0.83 (0.7)

Table B.12: Overall AUC (F-score) for **Wavelet plus SVM**

	svm					
	bpic2015_2	bpic2015_4	bpic2012_c	bpic2011_4	bpic2012_d	traffic
cluster_waveletAgg	0.54 (0.04)	0.54 (0.04)	0.62 (0.04)	0.58 (0.04)	0.57 (0.04)	0.7 (0.7)
prefix_WaveletIndex	0.6 (0.0)	0.54 (0.0)	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)
prefix_waveletLast	0.56 (0.01)	0.51 (0.01)	0.52 (0.01)	0.55 (0.01)	0.51 (0.01)	0.66 (0.67)
state_waveletAgg	0.74 (0.34)	0.57 (0.04)	0.64 (0.17)	0.53 (0.04)	0.56 (0.13)	0.66 (0.48)
single_waveletLast	0.52 (0.02)	0.52 (0.02)	0.52 (0.02)	0.62 (0.02)	0.52 (0.02)	0.61 (0.55)
single_waveletAgg	0.54 (0.04)	0.54 (0.04)	0.54 (0.04)	0.54 (0.04)	0.54 (0.04)	0.63 (0.54)
state_waveletLast	0.73 (0.32)	0.55 (0.02)	0.61 (0.15)	0.51 (0.11)	0.54 (0.11)	0.65 (0.55)
prefix_waveletAgg	0.52 (0.02)	0.51 (0.02)	0.52 (0.02)	0.54 (0.02)	0.52 (0.02)	0.54 (0.29)
cluster_waveletLast	0.76 (0.59)	0.52 (0.02)	0.6 (0.02)	0.77 (0.02)	0.53 (0.02)	0.67 (0.68)

	svm					
	bpic2012_a	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1
cluster_waveletAgg	0.52 (0.04)	0.53 (0.04)	0.73 (0.04)	0.55 (0.81)	0.54 (0.04)	0.68 (0.04)
prefix_WaveletIndex	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)	0.66 (0.8)	0.87 (0.0)	0.5 (0.0)
prefix_waveletLast	0.51 (0.31)	0.57 (0.11)	0.61 (0.01)	0.5 (0.78)	0.51 (0.01)	0.51 (0.01)
state_waveletAgg	0.64 (0.4)	0.53 (0.04)	0.72 (0.59)	0.59 (0.67)	0.37 (0.04)	0.66 (0.38)
single_waveletLast	0.54 (0.73)	0.52 (0.02)	0.61 (0.02)	0.52 (0.02)	0.52 (0.02)	0.54 (0.02)
single_waveletAgg	0.54 (0.75)	0.57 (0.12)	0.57 (0.04)	0.54 (0.81)	0.54 (0.04)	0.54 (0.04)
state_waveletLast	0.63 (0.35)	0.56 (0.05)	0.6 (0.02)	0.57 (0.65)	0.35 (0.02)	0.63 (0.36)
prefix_waveletAgg	0.52 (0.32)	0.56 (0.02)	0.52 (0.02)	0.52 (0.79)	0.52 (0.02)	0.52 (0.02)
cluster_waveletLast	0.6 (0.48)	0.57 (0.02)	0.84 (0.02)	0.52 (0.79)	0.52 (0.02)	0.48 (0.02)

	svm					
	bpic2015_5	sepsis_3	bpic2017_c	production	bpic2017_r	sepsis_1
cluster_waveletAgg	0.66 (0.32)	0.65 (0.31)	0.57 (0.34)	0.62 (0.34)	0.57 (0.04)	0.57 (0.07)
prefix_WaveletIndex	0.5 (0.0)	0.5 (0.0)	0.5 (0.09)	0.52 (0.0)	0.53 (0.01)	0.46 (0.0)
prefix_waveletLast	0.54 (0.01)	0.62 (0.01)	0.51 (0.01)	0.64 (0.54)	0.51 (0.01)	0.51 (0.01)
state_waveletAgg	0.54 (0.04)	0.56 (0.04)	0.68 (0.73)	0.7 (0.42)	0.57 (0.04)	0.55 (0.04)
single_waveletLast	0.64 (0.02)	0.71 (0.02)	**0.84** (**0.68**)	0.67 (0.58)	0.52 (0.02)	0.52 (0.02)
single_waveletAgg	0.54 (0.04)	0.59 (0.04)	0.67 (0.71)	0.65 (0.57)	0.54 (0.04)	0.53 (0.04)
state_waveletLast	0.7 (0.02)	0.61 (0.15)	0.81 (0.58)	0.63 (0.57)	0.52 (0.02)	0.52 (0.02)
prefix_waveletAgg	0.66 (0.02)	0.66 (0.02)	0.52 (0.02)	0.52 (0.02)	0.52 (0.02)	0.54 (0.02)
cluster_waveletLast	0.52 (0.02)	0.73 (0.02)	0.52 (0.02)	0.59 (0.13)	0.52 (0.02)	0.52 (0.1)

	svm	
	bpic2011_1	bpic2017_a
cluster_waveletAgg	0.53 (0.04)	0.49 (0.04)
prefix_WaveletIndex	0.5 (0.0)	0.5 (0.0)
prefix_waveletLast	0.52 (0.01)	0.51 (0.01)
state_waveletAgg	0.66 (0.09)	0.56 (0.15)
single_waveletLast	0.52 (0.02)	0.84 (0.68)
single_waveletAgg	0.59 (0.13)	0.67 (0.71)
state_waveletLast	0.64 (0.07)	0.54 (0.13)
prefix_waveletAgg	0.53 (0.47)	0.52 (0.02)
cluster_waveletLast	0.51 (0.35)	0.47 (0.02)

Table B.13: Execution times for **Wavelet plus XGBoost**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	893.4 ± 546.84	1440.93 ± 0.81	2084.13 ± 745.77	336.25 ± 0.17	570.53 ± 579.05	853.08 ± 0.41
prefix_waveletLast	12226.32 ± 588.4	788.32 ± 648.15	15446.7 ± 701.24	710.59 ± 781.44	8055.06 ± 366.87	924.25 ± 753.37
state_waveletAgg	$\mathbf{67.06 \pm 689.85}$	1080.62 ± 0.33	420.41 ± 485.59	526.75 ± 0.13	$\mathbf{71.52 \pm 575.01}$	1087.78 ± 0.32
single_waveletLast	5168.93 ± 616.23	3575.33 ± 1.02	8515.19 ± 746.83	445.32 ± 0.12	1565.68 ± 709.93	3701.11 ± 1.03
single_waveletAgg	1592.12 ± 466.31	2482.64 ± 0.96	6207.09 ± 623.6	376.72 ± 0.14	1221.17 ± 340.35	2540.4 ± 0.95
state_waveletLast	297.92 ± 345.62	4236.98 ± 1.17	$\mathbf{329.85 \pm 418.76}$	526.66 ± 0.11	242.23 ± 779.39	4251.3 ± 1.18
prefix_waveletAgg	157277.0 ± 650.57	$\mathbf{300.75 \pm 746.56}$	79441.32 ± 555.31	534.43 ± 654.0	121215.65 ± 646.38	$\mathbf{552.29 \pm 395.38}$
cluster_WaveletAgg	167501.35 ± 789.44	604.37 ± 0.6	1468810.91 ± 638.81	$\mathbf{177.77 \pm 0.19}$	112531.18 ± 619.59	629.06 ± 0.62
prefix_waveletIndex	6970.63 ± 725.9	534.9 ± 497.62	8809.01 ± 797.88	712.18 ± 500.94	4091.28 ± 537.26	556.06 ± 771.24

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	2332.56 ± 619.48	335.76 ± 0.17	26917.54 ± 568.5	682.85 ± 0.27	6159.88 ± 659.86	3490.7 ± 1.94
prefix_waveletLast	16307.62 ± 535.47	636.83 ± 579.95	44925.98 ± 654.3	746.02 ± 490.59	22239.55 ± 804.14	544.9 ± 669.69
state_waveletAgg	644.03 ± 501.9	526.64 ± 0.13	$\mathbf{2884.46 \pm 727.22}$	1126.79 ± 0.26	1303.48 ± 655.17	9250.65 ± 2.93
single_waveletLast	11150.93 ± 451.77	448.35 ± 0.12	114501.74 ± 440.5	1230.82 ± 0.27	9024.58 ± 444.59	5250.09 ± 1.7
single_waveletAgg	19625.99 ± 743.47	830.94 ± 0.31	86489.15 ± 589.5	1254.86 ± 0.37	3456.92 ± 581.43	3116.82 ± 1.25
state_waveletLast	$\mathbf{527.06 \pm 799.44}$	531.71 ± 0.12	7980.29 ± 383.41	1193.03 ± 0.24	$\mathbf{560.49 \pm 722.5}$	2738.48 ± 0.79
prefix_waveletAgg	80619.5 ± 432.46	550.66 ± 736.27	433272.14 ± 591.98	523.91 ± 595.01	68092.39 ± 646.16	$\mathbf{476.59 \pm 466.6}$
cluster_WaveletAgg	1232019.38 ± 398.24	$\mathbf{177.1 \pm 0.19}$	7796345.75 ± 364.77	447.28 ± 0.37	1392312.8 ± 605.2	2603.42 ± 2.89
prefix_waveletIndex	9153.18 ± 384.17	754.54 ± 597.02	59068.74 ± 407.37	$\mathbf{349.21 \pm 754.07}$	1952.92 ± 758.15	534.63 ± 739.14

	traffic		**bpic2012_A**		**bpic2015_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	42554.14 ± 750.51	2050.81 ± 0.37	5157.44 ± 557.89	342.27 ± 0.17	1492.19 ± 579.09	1297.42 ± 0.68
prefix_waveletLast	47141.64 ± 520.91	523.79 ± 506.3	15259.05 ± 497.67	459.29 ± 725.63	18064.39 ± 677.44	584.27 ± 336.44
state_waveletAgg	5764.92 ± 391.88	3707.2 ± 0.41	687.36 ± 345.84	518.66 ± 0.13	$\mathbf{151.51 \pm 731.76}$	1263.84 ± 0.38
single_waveletLast	51704.65 ± 588.98	3463.59 ± 0.35	8143.08 ± 489.37	445.78 ± 0.12	6475.84 ± 782.94	4199.14 ± 1.2
single_waveletAgg	71119.1 ± 506.51	2796.67 ± 0.37	2707.86 ± 705.0	365.53 ± 0.14	4074.23 ± 385.23	2889.5 ± 1.11
state_waveletLast	$\mathbf{4847.11 \pm 623.28}$	3896.81 ± 0.38	$\mathbf{523.08 \pm 674.97}$	513.43 ± 0.11	440.13 ± 637.4	4926.9 ± 1.37
prefix_waveletAgg	248414.66 ± 748.8	$\mathbf{305.43 \pm 374.09}$	81674.43 ± 635.31	569.41 ± 792.46	248893.1 ± 807.48	$\mathbf{348.82 \pm 674.02}$
cluster_WaveletAgg	7770732.32 ± 675.03	1088.08 ± 0.41	708732.63 ± 756.73	$\mathbf{164.93 \pm 0.17}$	504752.98 ± 490.92	706.73 ± 0.82
prefix_waveletIndex	226797.36 ± 645.39	809.34 ± 350.38	8897.46 ± 368.85	666.15 ± 573.83	7775.88 ± 708.57	410.7 ± 617.04

	bpic2011_3		**bpic2011_2**		**sepsis_2**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	2766.27 ± 476.02	5677.78 ± 3.37	2977.16 ± 536.26	3639.97 ± 1.95	218.13 ± 676.07	2186.48 ± 0.74
prefix_waveletLast	12952.65 ± 541.27	774.08 ± 630.19	24816.28 ± 486.65	761.3 ± 560.53	1552.6 ± 690.35	988.33 ± 732.43
state_waveletAgg	332.81 ± 545.47	11180.39 ± 2.93	799.78 ± 731.37	11421.53 ± 3.7	$\mathbf{40.42 \pm 386.09}$	3733.2 ± 0.71
single_waveletLast	2627.46 ± 739.61	4215.38 ± 1.13	3547.21 ± 651.56	3651.19 ± 1.1	248.11 ± 598.84	4223.26 ± 0.8
single_waveletAgg	1436.58 ± 474.61	2959.97 ± 0.97	2950.9 ± 492.88	2684.91 ± 1.04	505.27 ± 405.28	3794.91 ± 0.96
state_waveletLast	$\mathbf{194.1 \pm 774.17}$	3886.11 ± 0.92	$\mathbf{707.13 \pm 659.28}$	3561.25 ± 0.95	118.85 ± 561.96	8252.17 ± 2.24
prefix_waveletAgg	30242.76 ± 465.25	538.94 ± 408.95	55517.66 ± 478.24	$\mathbf{381.72 \pm 421.34}$	7598.81 ± 569.68	467.53 ± 367.43
cluster_WaveletAgg	1158564.91 ± 779.35	4614.44 ± 6.59	2508353.11 ± 387.77	2458.3 ± 2.42	31078.84 ± 387.53	715.44 ± 0.47
prefix_waveletIndex	2015.02 ± 588.34	$\mathbf{410.25 \pm 599.71}$	2075.84 ± 768.48	781.69 ± 603.08	1671.4 ± 705.84	$\mathbf{453.52 \pm 370.64}$

	bpic2015_1		**bpic2015_5**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	1264.62 ± 676.89	1870.58 ± 0.95	885.7 ± 413.55	1025.39 ± 0.52	684.75 ± 530.75	1903.2 ± 0.71
prefix_waveletLast	8555.6 ± 624.51	936.63 ± 778.03	15395.08 ± 740.14	981.68 ± 668.85	4444.67 ± 601.85	1001.68 ± 332.93
state_waveletAgg	$\mathbf{80.68 \pm 477.19}$	1339.92 ± 0.41	$\mathbf{102.94 \pm 347.8}$	1348.85 ± 0.4	$\mathbf{69.5 \pm 628.26}$	3130.26 ± 0.69
single_waveletLast	4160.23 ± 776.13	4512.28 ± 1.27	9142.05 ± 663.8	4564.01 ± 1.31	629.39 ± 519.07	7491.63 ± 2.21
single_waveletAgg	1945.4 ± 603.34	3097.65 ± 1.17	1989.24 ± 347.88	3438.52 ± 1.34	410.85 ± 532.13	4827.22 ± 1.43
state_waveletLast	336.77 ± 416.06	5191.71 ± 1.39	477.49 ± 750.6	5671.59 ± 1.5	78.51 ± 698.79	2662.54 ± 0.47
prefix_waveletAgg	150018.28 ± 416.04	441.88 ± 612.25	184672.99 ± 775.61	643.55 ± 357.35	23312.37 ± 614.95	$\mathbf{338.77 \pm 389.89}$
cluster_WaveletAgg	122690.07 ± 509.4	630.58 ± 0.61	441010.94 ± 351.53	1190.47 ± 1.42	76070.71 ± 569.73	610.97 ± 0.46
prefix_waveletIndex	4807.3 ± 640.94	$\mathbf{424.84 \pm 492.88}$	7606.26 ± 732.69	$\mathbf{628.89 \pm 427.7}$	1714.5 ± 576.27	711.68 ± 354.67

	bpic2017_C		**production**		**sepsis_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	174521.49 ± 619.06	1114.38 ± 0.43	28361.7 ± 550.29	3288.05 ± 1.05	472.51 ± 729.86	1654.43 ± 0.64
prefix_waveletLast	45625.28 ± 589.71	795.97 ± 577.23	1550.21 ± 654.98	808.64 ± 421.23	4265.27 ± 639.65	883.9 ± 681.38
state_waveletAgg	$\mathbf{3064.51 \pm 612.76}$	1174.51 ± 0.26	3708.37 ± 807.99	5669.88 ± 1.17	$\mathbf{44.42 \pm 557.42}$	2991.51 ± 0.66
single_waveletLast	150791.56 ± 383.62	1202.58 ± 0.26	687.42 ± 629.9	2284.38 ± 0.36	1206.06 ± 684.55	3138.35 ± 0.69
single_waveletAgg	28497.21 ± 482.55	967.66 ± 0.29	$\mathbf{195.41 \pm 770.85}$	3689.28 ± 0.84	420.28 ± 617.21	3421.14 ± 1.0
state_waveletLast	3991.49 ± 369.43	1200.11 ± 0.24	421.59 ± 616.49	2577.41 ± 0.39	185.91 ± 346.94	7910.19 ± 1.57
prefix_waveletAgg	238056.57 ± 674.45	453.39 ± 613.2	8641.38 ± 476.2	$\mathbf{357.62 \pm 506.14}$	22759.49 ± 671.02	$\mathbf{533.76 \pm 416.26}$
cluster_WaveletAgg	1853986.7 ± 809.36	592.99 ± 0.48	59878.53 ± 616.41	591.44 ± 0.37	81349.66 ± 721.93	649.01 ± 0.53
prefix_waveletIndex	59291.97 ± 374.4	$\mathbf{354.67 \pm 721.33}$	405.17 ± 713.31	787.4 ± 474.26	1609.56 ± 655.96	703.08 ± 538.74

	bpic2011_1		**bpic2017_R**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	5438.26 ± 384.57	3965.3 ± 1.95	15804.6 ± 506.57	682.88 ± 0.27
prefix_waveletLast	15889.92 ± 751.76	817.7 ± 367.79	45387.63 ± 488.07	590.48 ± 489.59
state_waveletAgg	837.38 ± 647.54	11612.09 ± 3.37	$\mathbf{1825.45 \pm 439.89}$	1119.55 ± 0.26
single_waveletLast	2866.95 ± 404.27	3883.21 ± 1.08	67442.43 ± 429.92	1211.73 ± 0.26
single_waveletAgg	2037.05 ± 439.26	2958.21 ± 1.04	23445.6 ± 715.12	1228.17 ± 0.36
state_waveletLast	$\mathbf{277.3 \pm 734.85}$	3669.49 ± 0.86	2581.03 ± 688.9	1195.76 ± 0.24
prefix_waveletAgg	40609.81 ± 620.7	610.05 ± 393.27	238160.52 ± 626.49	$\mathbf{347.72 \pm 767.08}$
cluster_WaveletAgg	914889.78 ± 582.67	6650.51 ± 8.49	17750217.43 ± 559.92	455.3 ± 0.37
prefix_waveletIndex	1893.52 ± 546.51	$\mathbf{477.64 \pm 698.8}$	61009.36 ± 628.06	559.64 ± 608.82

Table B.14: Execution times for **Wavelet plus RF**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	9780.89 ± 3683.28	30532.11 ± 26.83	3045.72 ± 5031.2	7762.52 ± 3.87	4707.33 ± 5881.2	23804.94 ± 13.12
prefix_waveletLast	201237.99 ± 2667.52	5259.74 ± 2555.53	114956.42 ± 5256.03	5911.05 ± 2577.98	142632.4 ± 4774.15	7480.02 ± 3201.23
state_waveletAgg	**3202.09 ± 4122.02**	17000.57 ± 5.36	**1857.04 ± 4318.7**	12344.39 ± 3.54	**2643.09 ± 3979.29**	17594.05 ± 5.48
single_waveletLast	8731.36 ± 4754.81	58769.59 ± 17.51	9888.15 ± 4101.79	20400.6 ± 5.62	7216.71 ± 3435.0	59612.02 ± 16.66
single_waveletAgg	5259.91 ± 4963.26	46800.06 ± 18.71	6731.42 ± 4883.61	9953.45 ± 3.59	4030.9 ± 4265.1	42522.41 ± 15.95
state_waveletLast	17939.32 ± 5251.52	42854.56 ± 11.32	4555.37 ± 5593.79	17347.78 ± 4.48	10724.04 ± 5254.55	28167.45 ± 7.41
prefix_waveletAgg	1032470.95 ± 5137.54	**4701.69 ± 6127.01**	590180.42 ± 3536.68	**3759.71 ± 3456.11**	744373.02 ± 4297.33	**3412.56 ± 4553.93**
cluster_WaveletAgg	3529746.98 ± 4178.68	25234.41 ± 27.32	1187041.54 ± 4489.06	3876.85 ± 3.96	1167820.02 ± 3203.48	17132.86 ± 17.93
prefix_waveletIndex	34452.5 ± 4398.49	5892.99 ± 4300.96	69564.82 ± 3505.8	4287.05 ± 2638.86	24671.41 ± 3045.45	3450.82 ± 4552.09

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	5474.3 ± 3169.12	10927.75 ± 5.55	11580.44 ± 4444.47	12440.34 ± 4.96	12937.14 ± 2955.28	37169.14 ± 20.89
prefix_waveletLast	119552.62 ± 3021.79	4904.47 ± 5968.24	350619.03 ± 2635.5	4514.16 ± 4381.48	142215.17 ± 2780.08	**3482.07 ± 3427.71**
state_waveletAgg	**1369.8 ± 3250.58**	12136.97 ± 3.59	4039.0 ± 4362.76	11023.24 ± 2.52	19176.71 ± 5383.37	100214.77 ± 30.65
single_waveletLast	12172.43 ± 4610.24	19906.38 ± 5.84	25723.96 ± 4180.89	11629.54 ± 2.54	**8765.31 ± 4819.5**	39696.67 ± 11.32
single_waveletAgg	3459.55 ± 4143.79	9850.72 ± 3.55	34407.41 ± 4771.34	9675.33 ± 2.85	11709.82 ± 5240.36	110643.26 ± 42.72
state_waveletLast	4501.48 ± 5058.84	17845.61 ± 5.61	**3032.5 ± 3687.19**	9016.01 ± 1.84	24578.03 ± 4099.36	103825.13 ± 28.36
prefix_waveletAgg	613842.47 ± 3548.07	**2456.22 ± 5159.4**	1810148.59 ± 3235.24	**2533.47 ± 2670.35**	437128.41 ± 3792.12	4851.09 ± 6113.2
cluster_WaveletAgg	4792546.74 ± 3845.57	5072.74 ± 6.14	8957470.91 ± 3269.72	3488.81 ± 2.85	786710.15 ± 4476.85	8675.34 ± 9.06
prefix_waveletIndex	74357.18 ± 4063.15	4075.65 ± 3360.68	448288.27 ± 5418.28	4602.38 ± 4994.28	15747.4 ± 2904.93	4711.69 ± 3359.91

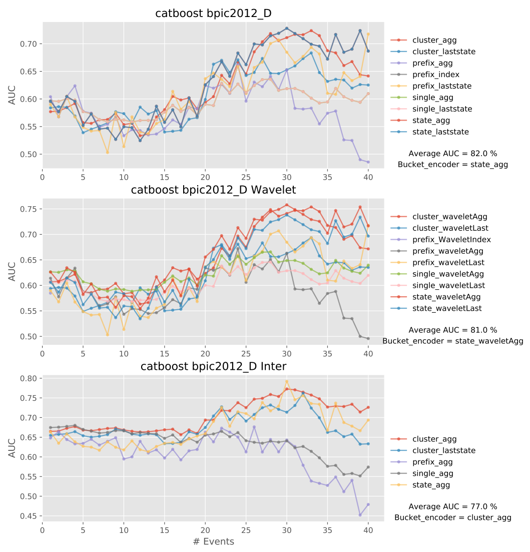
	traffic		**bpic2012_A**		**bpic2015_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	8964.1 ± 5544.32	14880.71 ± 2.8	3322.16 ± 4663.34	7611.92 ± 3.77	42695.89 ± 4833.43	40233.42 ± 42.27
prefix_waveletLast	365187.7 ± 5505.34	**3628.08 ± 4404.39**	117044.05 ± 5971.82	4453.61 ± 5900.12	339093.81 ± 4863.61	4436.38 ± 2544.74
state_waveletAgg	**1162.18 ± 2888.79**	27465.46 ± 3.01	**1342.57 ± 4692.72**	12104.6 ± 3.47	**6209.1 ± 5940.83**	17911.64 ± 5.57
single_waveletLast	7066.49 ± 6071.58	25114.9 ± 2.52	5673.72 ± 6010.76	12606.58 ± 3.45	10860.77 ± 4420.01	46412.35 ± 13.31
single_waveletAgg	5986.14 ± 4083.33	20759.84 ± 2.73	64083.1 ± 2808.5	10074.81 ± 3.67	9161.7 ± 3023.85	44872.56 ± 17.74
state_waveletLast	1670.63 ± 2854.66	29218.84 ± 2.85	4283.35 ± 4665.23	16279.75 ± 4.62	15609.63 ± 5915.29	43535.15 ± 14.34
prefix_waveletAgg	1900878.0 ± 4695.82	4286.45 ± 4872.95	629696.45 ± 3051.78	4784.4 ± 2634.71	1619124.34 ± 6040.12	**4039.4 ± 6052.24**
cluster_WaveletAgg	1745573.11 ± 2678.86	8091.15 ± 2.99	1359822.31 ± 4819.76	**3772.31 ± 3.81**	6451753.73 ± 4918.67	29805.65 ± 30.98
prefix_waveletIndex	1725048.79 ± 4802.22	5024.83 ± 5830.32	67789.59 ± 4497.14	5841.87 ± 5319.53	58302.06 ± 5800.26	5219.58 ± 3352.76

	bpic2011_3		**bpic2011_2**		**sepsis_2**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	**2679.21 ± 5213.65**	56114.93 ± 29.4	10033.14 ± 5975.31	45624.69 ± 24.31	1630.66 ± 4876.78	22519.47 ± 7.82
prefix_waveletLast	67785.58 ± 4674.73	7835.44 ± 4906.94	133569.09 ± 5585.96	**3541.63 ± 5632.54**	10849.14 ± 2928.95	8061.99 ± 3703.45
state_waveletAgg	5373.73 ± 3996.05	92175.14 ± 24.88	11091.02 ± 2946.21	81390.5 ± 25.16	**881.26 ± 3571.97**	68803.69 ± 19.36
single_waveletLast	3680.52 ± 4437.2	49122.74 ± 12.17	**2479.35 ± 5495.76**	31997.19 ± 9.23	4485.75 ± 5877.3	125090.9 ± 23.27
single_waveletAgg	4600.09 ± 2888.43	150556.16 ± 50.17	4458.92 ± 2666.91	106377.11 ± 40.2	4650.59 ± 5041.79	102684.56 ± 24.01
state_waveletLast	14623.8 ± 6067.54	118268.44 ± 27.79	15329.76 ± 4875.62	86806.34 ± 22.93	1529.16 ± 4608.51	65579.91 ± 9.62
prefix_waveletAgg	217035.55 ± 5586.97	**4752.27 ± 5082.66**	419791.82 ± 4572.07	4585.36 ± 4115.85	57580.11 ± 3310.89	**4639.77 ± 3852.52**
cluster_WaveletAgg	572147.58 ± 4662.49	10436.41 ± 9.62	1055063.36 ± 3070.21	10829.72 ± 11.32	906544.85 ± 4319.59	15227.93 ± 9.88
prefix_waveletIndex	15757.35 ± 2922.56	5457.84 ± 3964.36	15684.8 ± 5492.77	4504.79 ± 4134.56	13299.72 ± 5469.38	4846.91 ± 3892.31

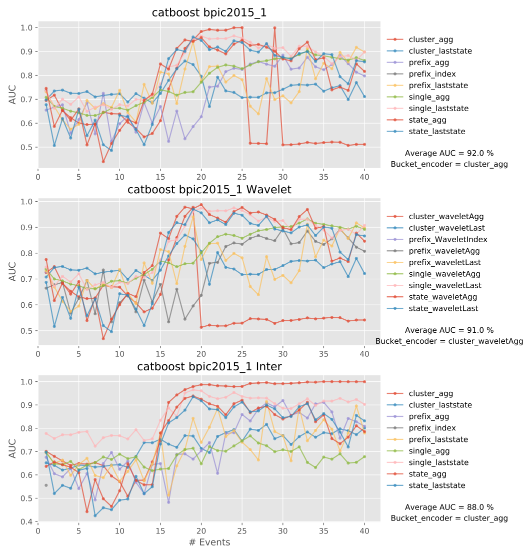
	bpic2015_1		**bpic2015_5**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	5900.51 ± 3116.26	34087.82 ± 19.03	8992.24 ± 5295.81	27049.5 ± 14.8	2641.05 ± 3001.03	27319.66 ± 9.97
prefix_waveletLast	230358.89 ± 4846.0	6007.68 ± 4346.51	260568.24 ± 5611.79	5076.57 ± 5828.69	32707.43 ± 3971.59	6380.46 ± 6139.51
state_waveletAgg	**3686.7 ± 4461.36**	19339.6 ± 6.01	**6554.98 ± 5304.07**	18452.47 ± 5.85	**1372.22 ± 5714.01**	62842.37 ± 10.13
single_waveletLast	6727.64 ± 2870.59	69907.27 ± 20.05	17524.07 ± 3509.86	64265.06 ± 18.43	4267.63 ± 5574.77	135720.9 ± 31.36
single_waveletAgg	3760.74 ± 3225.95	45965.77 ± 18.19	8456.84 ± 3507.25	50244.92 ± 19.71	4039.91 ± 2970.93	69067.94 ± 19.84
state_waveletLast	17690.37 ± 5807.52	41540.42 ± 11.78	36716.52 ± 4108.53	52265.47 ± 14.01	2706.66 ± 4541.74	86516.48 ± 16.97
prefix_waveletAgg	965938.37 ± 5448.93	**4701.55 ± 3165.19**	1381767.43 ± 5667.78	3920.57 ± 2695.45	181859.22 ± 3669.31	**3992.36 ± 5360.02**
cluster_WaveletAgg	4573035.58 ± 5065.91	37471.45 ± 38.24	4927709.83 ± 4528.04	26090.6 ± 27.9	1089688.53 ± 4562.82	13105.95 ± 9.8
prefix_waveletIndex	31511.13 ± 5994.85	5326.96 ± 4770.01	47294.85 ± 2856.01	**3056.47 ± 3312.53**	12977.96 ± 3819.3	4578.09 ± 5161.85

	bpic2017_C		**production**		**sepsis_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	25867.5 ± 2577.6	16867.81 ± 7.36	3429.32 ± 4403.65	27765.28 ± 8.17	3752.49 ± 2615.17	30360.31 ± 11.49
prefix_waveletLast	354345.45 ± 3091.62	7540.18 ± 4813.31	26766.86 ± 2901.92	7682.99 ± 3977.07	33535.07 ± 5944.43	7483.63 ± 3256.08
state_waveletAgg	**1930.25 ± 3499.03**	8616.26 ± 1.95	1406.66 ± 5553.55	134295.42 ± 25.41	1909.33 ± 3271.96	115976.73 ± 28.71
single_waveletLast	1146746.17 ± 5468.27	9145.42 ± 2.0	1392.03 ± 3663.48	40712.48 ± 6.26	6361.4 ± 5152.65	125301.73 ± 27.72
single_waveletAgg	216716.85 ± 4868.89	7358.92 ± 2.17	**391.47 ± 5960.15**	32476.49 ± 6.51	**750.39 ± 3414.31**	40499.28 ± 11.64
state_waveletLast	5542.97 ± 3688.54	23174.17 ± 4.98	3226.72 ± 6043.02	83020.3 ± 14.28	855.75 ± 4607.28	35457.22 ± 9.11
prefix_waveletAgg	1810382.92 ± 3437.04	**2808.34 ± 5228.66**	89618.2 ± 4338.06	**3697.02 ± 5830.0**	170857.89 ± 4662.51	**3442.7 ± 4537.02**
cluster_WaveletAgg	83067902.14 ± 5050.5	2996.34 ± 2.47	1819916.81 ± 4862.34	14779.46 ± 8.43	417485.4 ± 5584.58	8728.7 ± 6.72
prefix_waveletIndex	453142.77 ± 4489.77	6103.75 ± 5434.79	2980.54 ± 4185.23	4128.02 ± 5110.42	14278.04 ± 3084.84	4042.86 ± 5096.96

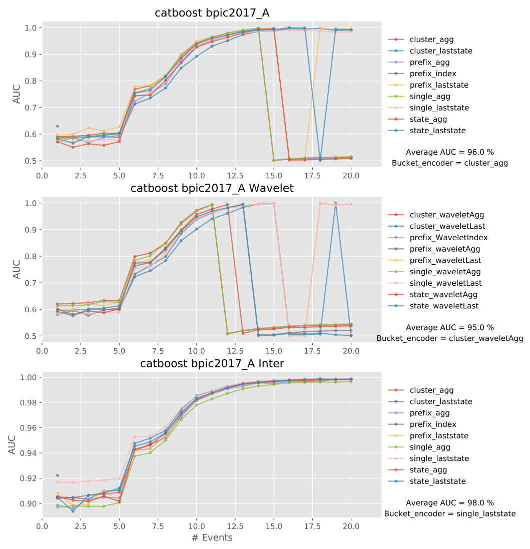
	bpic2011_1		**bpic2017_R**			
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)		
cluster_WaveletLast	11536.44 ± 5607.7	59931.83 ± 28.12	14592.49 ± 4954.31	12718.6 ± 5.07		
prefix_waveletLast	107650.3 ± 5304.89	6947.5 ± 6032.15	345157.81 ± 5388.84	8147.58 ± 2895.61		
state_waveletAgg	14291.37 ± 3957.72	96843.99 ± 27.72	7667.6 ± 4702.18	22248.8 ± 5.19		
single_waveletLast	5773.57 ± 5048.9	58874.84 ± 16.75	44204.62 ± 2573.84	22633.93 ± 5.0		
single_waveletAgg	7587.2 ± 4800.44	141931.72 ± 50.79	35202.22 ± 3537.57	28730.4 ± 8.24		
state_waveletLast	**5219.78 ± 4397.87**	35443.42 ± 8.56	**5283.77 ± 3447.29**	34559.23 ± 7.4		
prefix_waveletAgg	385110.12 ± 4011.33	**2415.9 ± 2839.53**	1819286.41 ± 2679.78	**3841.77 ± 4827.17**		
cluster_WaveletAgg	583685.64 ± 5436.92	9690.73 ± 9.44	3909169.66 ± 2559.28	6291.55 ± 5.12		
prefix_waveletIndex	14444.08 ± 4400.42	2653.92 ± 6134.95	456518.14 ± 5778.13	5417.12 ± 3042.18		



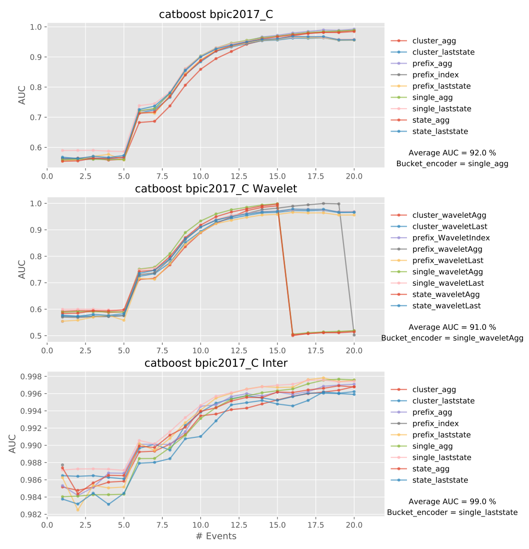
(a) Bpic2012_D



(b) Bpic2015

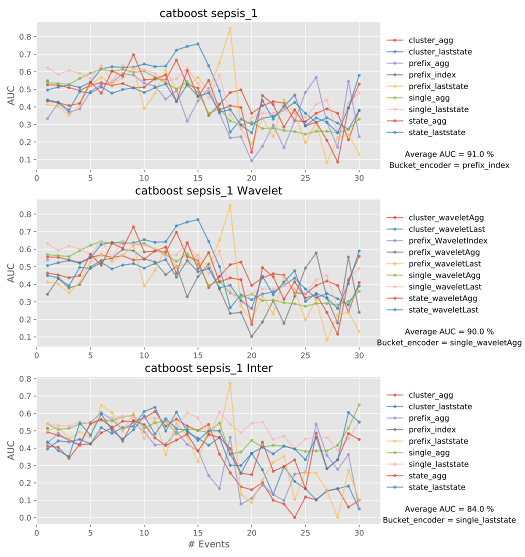


(c) Bpic2017_A

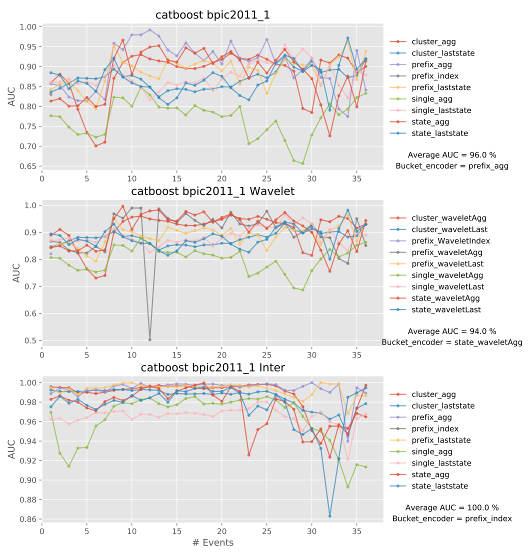


(d) Bpic2017_C

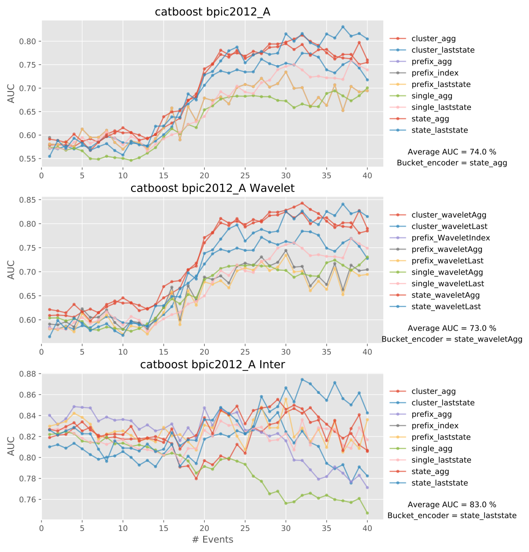
Figure B.22: Comparing CatBoost before and after adding Inter-case features on all event logs



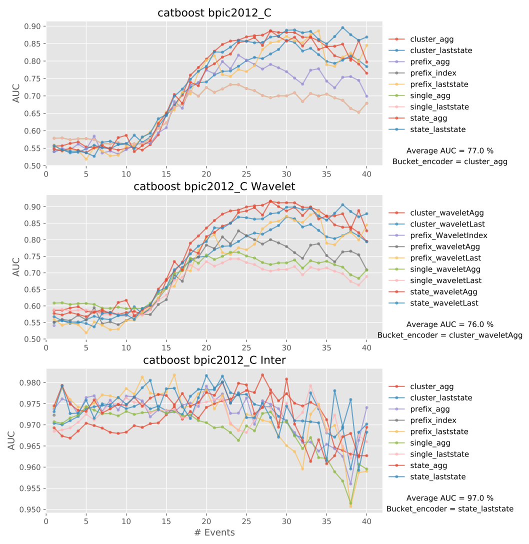
(a) Sepsis_1



(b) Bpic2011

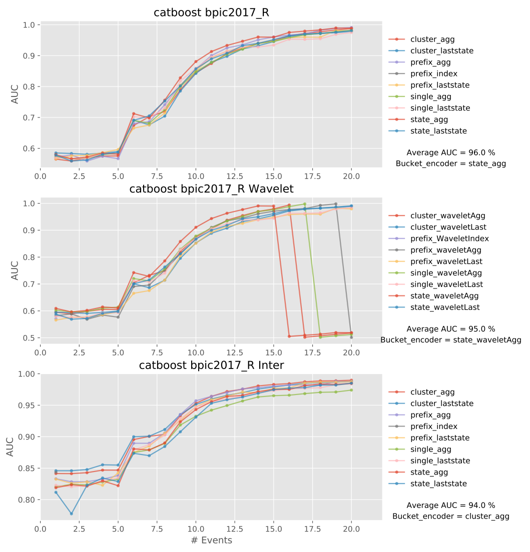


(c) Bpic2012_A

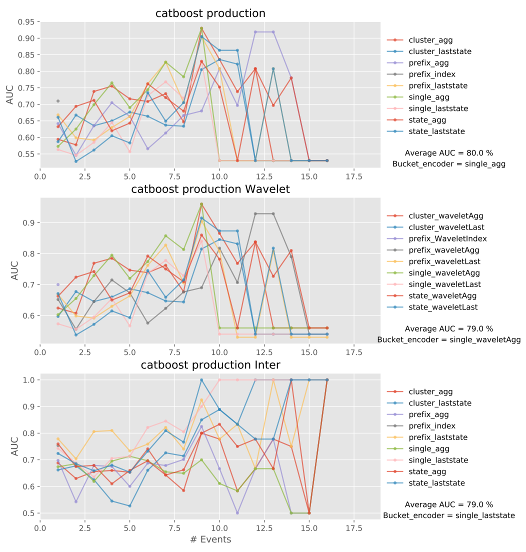


(d) Bpic2012_C

Figure B.23: Comparing CatBoost before and after adding Inter-case features on all event logs (*continued*).



(a) Wavelet bpic2017_R



(b) Production

Figure B.24: Comparing CatBoost before and after adding Inter-case features on all event logs (*continued*)

Table B.15: Execution times for **Wavelet plus Logit**

	bpic2015_2		bpic2012_C		bpic2015_4	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	208.2 ± 242.86	675.7 ± 0.37	209.08 ± 309.36	190.13 ± 0.1	197.73 ± 239.62	715.76 ± 0.35
prefix_waveletLast	4961.0 ± 293.05	360.43 ± 345.88	8563.18 ± 267.53	490.1 ± 275.0	3993.51 ± 357.56	254.23 ± 432.23
state_waveletAgg	**42.27 ± 400.51**	1244.23 ± 0.35	**39.11 ± 292.89**	289.74 ± 0.07	**26.14 ± 232.71**	718.19 ± 0.2
single_waveletLast	204.29 ± 421.48	587.39 ± 0.17	282.43 ± 445.15	250.51 ± 0.07	102.56 ± 339.33	1042.02 ± 0.33
single_waveletAgg	282.17 ± 412.59	1601.08 ± 0.64	140.16 ± 355.1	204.75 ± 0.08	151.02 ± 267.05	1588.06 ± 0.62
state_waveletLast	207.34 ± 379.9	2956.73 ± 0.9	61.84 ± 390.46	289.6 ± 0.06	127.77 ± 307.99	2418.67 ± 0.66
prefix_waveletAgg	85664.97 ± 318.84	**236.79 ± 272.1**	45403.09 ± 225.38	357.15 ± 259.54	61874.45 ± 209.84	301.88 ± 331.4
cluster_WaveletAgg	53676.55 ± 220.73	470.72 ± 0.55	42301.37 ± 238.69	**98.13 ± 0.1**	39753.93 ± 284.9	349.57 ± 0.34
prefix_waveletIndex	2797.33 ± 234.25	287.09 ± 329.7	4965.72 ± 192.39	339.84 ± 315.37	2306.9 ± 213.19	**211.27 ± 378.08**

	bpic2012_D		bpic2017_A		bpic2011_4	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	155.79 ± 304.62	191.35 ± 0.1	822.77 ± 325.14	372.74 ± 0.15	748.19 ± 381.12	2041.17 ± 1.1
prefix_waveletLast	8558.63 ± 187.76	452.67 ± 435.83	25613.72 ± 194.54	312.89 ± 199.69	14175.27 ± 313.68	454.48 ± 286.34
state_waveletAgg	**44.3 ± 386.68**	301.18 ± 0.07	295.06 ± 219.09	805.28 ± 0.18	151.68 ± 209.42	4730.1 ± 1.46
single_waveletLast	169.16 ± 223.29	247.13 ± 0.07	1879.21 ± 239.33	849.57 ± 0.19	143.07 ± 274.21	1472.74 ± 0.42
single_waveletAgg	85.5 ± 381.54	212.39 ± 0.08	2513.56 ± 273.49	706.81 ± 0.21	416.49 ± 382.67	2768.07 ± 1.13
state_waveletLast	77.53 ± 438.77	300.46 ± 0.06	**221.53 ± 417.79**	658.65 ± 0.13	**54.02 ± 447.89**	1330.8 ± 0.35
prefix_waveletAgg	45803.35 ± 432.36	184.52 ± 217.89	132559.16 ± 344.89	**192.76 ± 264.07**	31889.52 ± 240.92	**161.85 ± 321.54**
cluster_WaveletAgg	51632.99 ± 355.42	**99.8 ± 0.11**	654368.87 ± 423.17	254.87 ± 0.21	187617.89 ± 434.74	791.54 ± 0.77
prefix_waveletIndex	4984.35 ± 265.02	356.83 ± 423.14	32925.94 ± 263.68	340.08 ± 303.59	1101.71 ± 328.15	393.62 ± 407.93

	traffic		bpic2012_A		bpic2015_3	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	654.85 ± 236.14	1087.08 ± 0.2	257.96 ± 204.07	187.03 ± 0.1	549.32 ± 189.13	860.51 ± 0.44
prefix_waveletLast	26678.01 ± 305.67	597.87 ± 312.19	8380.7 ± 368.29	501.95 ± 352.52	8478.16 ± 321.74	586.57 ± 404.48
state_waveletAgg	**84.9 ± 440.19**	2006.43 ± 0.22	746.67 ± 233.89	292.31 ± 0.07	**117.96 ± 398.21**	1955.06 ± 0.44
single_waveletLast	516.23 ± 280.98	1834.72 ± 0.18	286.84 ± 319.16	254.35 ± 0.07	147.92 ± 394.89	717.15 ± 0.21
single_waveletAgg	437.3 ± 439.82	1516.57 ± 0.2	**142.31 ± 336.04**	204.28 ± 0.08	749.8 ± 368.02	1975.64 ± 0.79
state_waveletLast	122.04 ± 200.74	2134.52 ± 0.21	780.08 ± 371.43	293.92 ± 0.06	392.25 ± 378.49	3469.12 ± 0.96
prefix_waveletAgg	138864.57 ± 322.91	**277.24 ± 403.55**	44235.45 ± 267.59	331.27 ± 219.91	127704.94 ± 225.15	**214.03 ± 184.51**
cluster_WaveletAgg	127519.11 ± 348.59	591.08 ± 0.22	53843.21 ± 242.36	**98.26 ± 0.1**	114763.79 ± 441.36	344.64 ± 0.34
prefix_waveletIndex	126019.75 ± 360.79	356.06 ± 405.86	4998.5 ± 343.72	223.03 ± 212.94	4357.62 ± 425.84	373.94 ± 258.29

	bpic2011_3		bpic2011_2		sepsis_2	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	238.25 ± 313.01	2293.67 ± 1.18	260.35 ± 305.88	987.99 ± 0.5	69.76 ± 206.42	1153.57 ± 0.4
prefix_waveletLast	7724.09 ± 211.08	576.03 ± 260.15	13664.77 ± 332.74	414.34 ± 375.3	802.37 ± 380.35	**265.72 ± 299.13**
state_waveletAgg	95.22 ± 241.92	5180.68 ± 1.38	175.51 ± 260.6	5415.67 ± 1.73	**30.72 ± 382.46**	4577.0 ± 1.2
single_waveletLast	157.19 ± 377.98	2130.44 ± 0.54	298.78 ± 259.82	2235.18 ± 0.66	66.0 ± 277.93	2270.56 ± 0.45
single_waveletAgg	140.85 ± 284.71	2163.83 ± 0.74	259.88 ± 300.17	1941.4 ± 0.79	76.22 ± 419.26	2806.43 ± 0.76
state_waveletLast	**37.25 ± 394.61**	1651.23 ± 0.39	**80.46 ± 444.29**	1683.58 ± 0.46	48.39 ± 259.4	5240.44 ± 1.06
prefix_waveletAgg	14862.77 ± 395.42	**320.43 ± 441.08**	31083.19 ± 290.78	**216.8 ± 234.11**	4295.18 ± 320.15	350.42 ± 221.96
cluster_WaveletAgg	95977.07 ± 439.67	1956.75 ± 2.35	98225.87 ± 188.85	644.39 ± 0.47	6801.36 ± 323.95	392.31 ± 0.25
prefix_waveletIndex	1028.62 ± 401.94	411.41 ± 380.17	1166.4 ± 368.87	433.12 ± 400.8	991.1 ± 325.44	417.02 ± 253.79

	bpic2015_1		bpic2015_5		sepsis_3	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	239.86 ± 229.34	710.93 ± 0.35	307.36 ± 297.99	599.11 ± 0.31	107.36 ± 360.09	918.2 ± 0.34
prefix_waveletLast	4755.58 ± 276.34	538.82 ± 447.06	6501.54 ± 405.17	515.21 ± 415.34	2424.75 ± 231.96	596.42 ± 390.16
state_waveletAgg	**38.53 ± 428.1**	922.35 ± 0.27	**28.73 ± 234.0**	739.14 ± 0.22	**13.97 ± 448.68**	1801.99 ± 0.4
single_waveletLast	103.73 ± 428.11	760.47 ± 0.22	228.6 ± 291.73	739.91 ± 0.21	140.67 ± 271.61	1686.96 ± 0.38
single_waveletAgg	269.16 ± 399.27	2050.64 ± 0.79	256.05 ± 253.99	1705.05 ± 0.65	102.44 ± 208.98	3689.2 ± 1.04
state_waveletLast	296.9 ± 336.56	3614.31 ± 1.06	212.27 ± 207.83	2937.86 ± 0.82	29.32 ± 236.12	2080.98 ± 0.32
prefix_waveletAgg	70290.43 ± 205.92	319.74 ± 438.05	106468.21 ± 281.32	**197.16 ± 195.42**	12770.4 ± 186.26	**329.54 ± 392.18**
cluster_WaveletAgg	54897.89 ± 373.45	343.95 ± 0.37	131334.94 ± 251.98	546.6 ± 0.55	13738.56 ± 289.82	371.17 ± 0.31
prefix_waveletIndex	3877.99 ± 377.39	**270.98 ± 279.63**	5540.75 ± 280.9	322.42 ± 299.93	916.16 ± 314.23	355.16 ± 273.91

	bpic2017_C		production		sepsis_1	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	869.71 ± 383.17	375.76 ± 0.15	111.12 ± 227.0	978.02 ± 0.3	139.44 ± 374.68	1052.29 ± 0.43
prefix_waveletLast	25675.84 ± 186.71	380.6 ± 374.37	1818.58 ± 303.27	528.29 ± 412.16	2456.92 ± 192.0	521.3 ± 365.0
state_waveletAgg	**141.01 ± 226.83**	629.44 ± 0.14	**20.1 ± 390.45**	2295.05 ± 0.41	**49.28 ± 188.28**	4430.0 ± 0.98
single_waveletLast	83773.09 ± 424.92	668.1 ± 0.15	47.51 ± 340.89	1458.16 ± 0.26	125.09 ± 268.19	1442.08 ± 0.32
single_waveletAgg	15831.79 ± 272.36	537.59 ± 0.16	23.88 ± 259.53	992.37 ± 0.2	101.1 ± 238.21	2853.55 ± 0.83
state_waveletLast	206.09 ± 336.36	661.25 ± 0.13	25.5 ± 429.62	1712.39 ± 0.32	83.83 ± 282.99	3541.12 ± 0.78
prefix_waveletAgg	132253.65 ± 261.59	252.48 ± 443.41	13440.67 ± 420.58	**160.84 ± 369.5**	12637.84 ± 348.73	253.67 ± 292.45
cluster_WaveletAgg	6068347.83 ± 356.99	**218.89 ± 0.18**	28012.22 ± 307.77	353.6 ± 0.25	19594.48 ± 437.64	319.5 ± 0.25
prefix_waveletIndex	32766.5 ± 317.88	437.51 ± 258.24	213.47 ± 220.21	387.32 ± 237.33	1144.49 ± 349.88	**203.37 ± 371.94**

	bpic2011_1		bpic2017_R	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	179.79 ± 408.03	989.86 ± 0.5	886.6 ± 295.36	389.59 ± 0.15
prefix_waveletLast	10091.9 ± 401.03	336.65 ± 316.35	25423.62 ± 416.98	307.81 ± 186.9
state_waveletAgg	149.36 ± 316.58	6083.99 ± 1.76	**139.6 ± 325.47**	634.31 ± 0.15
single_waveletLast	248.84 ± 365.36	3320.96 ± 0.88	1547.55 ± 249.95	672.84 ± 0.15
single_waveletAgg	165.18 ± 418.09	1830.09 ± 0.64	3121.27 ± 419.03	697.88 ± 0.21
state_waveletLast	**98.16 ± 384.69**	2217.66 ± 0.82	226.33 ± 224.84	655.5 ± 0.13
prefix_waveletAgg	21171.32 ± 389.99	**186.85 ± 449.06**	134165.63 ± 317.95	307.59 ± 305.21
cluster_WaveletAgg	121013.34 ± 338.98	1719.05 ± 1.97	257450.51 ± 276.83	**190.88 ± 0.16**
prefix_waveletIndex	1043.2 ± 424.03	348.79 ± 335.92	32794.73 ± 403.0	272.36 ± 412.03

Table B.16: Execution times for **Wavelet plus SVM**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	143.3 ± 347.57	444.21 ± 0.23	522.08 ± 433.76	**168.11 ± 0.08**	121.78 ± 298.81	489.97 ± 0.24
prefix_waveletLast	12054.2 ± 246.05	590.06 ± 186.36	8553.14 ± 266.04	346.18 ± 190.56	8999.25 ± 327.07	278.12 ± 226.37
state_waveletAgg	**18.69 ± 409.19**	592.61 ± 0.18	39.78 ± 427.99	291.56 ± 0.07	**16.99 ± 378.75**	638.27 ± 0.2
single_waveletLast	1010.57 ± 240.52	2493.15 ± 0.73	14779.68 ± 362.95	281.36 ± 0.08	595.5 ± 364.33	3160.55 ± 0.92
single_waveletAgg	722.42 ± 208.75	1430.38 ± 0.56	1296.53 ± 233.1	213.47 ± 0.08	174.85 ± 384.11	1549.14 ± 0.58
state_waveletLast	113.63 ± 340.76	2175.03 ± 0.6	**35.75 ± 247.11**	295.81 ± 0.07	82.18 ± 366.41	1901.6 ± 0.53
prefix_waveletAgg	78552.12 ± 192.07	**352.19 ± 221.53**	44720.58 ± 373.32	227.49 ± 278.55	59573.5 ± 227.69	**225.76 ± 288.54**
cluster_WaveletAgg	1849173.63 ± 19437.02	12333.97 ± 13.28	25535404.74 ± 8798.71	4112.38 ± 4.17	1823097.61 ± 15065.2	16655.56 ± 19.72
prefix_waveletIndex	2992.93 ± 427.6	362.83 ± 212.92	4848.66 ± 215.4	217.3 ± 373.93	1885.51 ± 281.42	396.41 ± 393.3

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	2140.58 ± 224.38	**193.63 ± 0.1**	25949.28 ± 372.09	426.87 ± 0.17	1445.48 ± 249.02	2967.24 ± 1.63
prefix_waveletLast	8583.65 ± 369.15	456.32 ± 192.16	25031.09 ± 368.16	512.16 ± 361.12	8760.33 ± 432.78	406.57 ± 325.28
state_waveletAgg	**39.91 ± 248.57**	292.77 ± 0.07	**295.06 ± 218.75**	805.28 ± 0.18	116.9 ± 193.42	5793.07 ± 1.86
single_waveletLast	3064.4 ± 277.74	261.37 ± 0.07	224245.94 ± 322.22	931.76 ± 0.2	4061.38 ± 298.82	2758.96 ± 0.84
single_waveletAgg	1824.09 ± 242.03	214.86 ± 0.08	2513.56 ± 407.55	706.81 ± 0.21	7153.56 ± 358.22	8301.01 ± 3.19
state_waveletLast	48.18 ± 283.92	287.78 ± 0.06	2923.51 ± 350.91	859.42 ± 0.17	**51.75 ± 423.73**	1580.37 ± 0.44
prefix_waveletAgg	44582.97 ± 307.97	378.72 ± 436.91	132236.53 ± 440.07	**250.37 ± 213.84**	46378.81 ± 248.17	**239.6 ± 282.43**
cluster_WaveletAgg	30428074.28 ± 13935.32	4341.12 ± 4.53	29650664.03 ± 15501.1	11548.53 ± 9.45	4486206.16 ± 11335.8	18357.8 ± 19.31
prefix_waveletIndex	4911.62 ± 407.88	306.64 ± 399.15	32781.76 ± 445.16	320.36 ± 342.89	1111.45 ± 285.87	410.63 ± 232.57

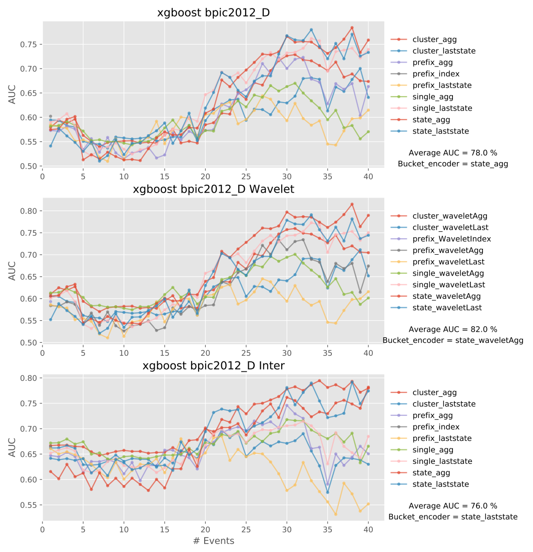
	traffic		**bpic2012_A**		**bpic2015_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	23641.19 ± 226.99	1139.34 ± 0.21	955.57 ± 294.67	**192.04 ± 0.1**	295.61 ± 448.63	572.49 ± 0.28
prefix_waveletLast	26189.8 ± 210.53	527.17 ± 321.08	8459.39 ± 194.49	319.93 ± 380.75	17102.65 ± 445.75	293.71 ± 415.78
state_waveletAgg	**1111.16 ± 408.06**	2100.4 ± 0.24	**33.26 ± 416.66**	299.22 ± 0.07	**27.85 ± 401.99**	705.55 ± 0.21
single_waveletLast	28724.81 ± 418.07	1924.22 ± 0.19	4895.46 ± 252.99	274.22 ± 0.07	4057.52 ± 313.23	4176.01 ± 1.27
single_waveletAgg	39510.61 ± 407.32	1553.71 ± 0.2	3440.91 ± 407.04	231.97 ± 0.09	771.75 ± 438.69	1672.82 ± 0.64
state_waveletLast	1969.63 ± 214.63	2281.14 ± 0.23	58.82 ± 354.71	295.14 ± 0.06	129.98 ± 188.37	2370.92 ± 0.72
prefix_waveletAgg	137085.21 ± 255.28	354.05 ± 251.11	45385.78 ± 244.8	221.55 ± 219.29	125975.67 ± 286.44	**291.33 ± 449.93**
cluster_WaveletAgg	195614587.33 ± 10299.18	27390.6 ± 10.2	19039026.29 ± 16642.15	4694.18 ± 4.96	5955365.53 ± 14387.68	14655.94 ± 15.51
prefix_waveletIndex	128330.25 ± 422.69	**348.06 ± 388.77**	4842.24 ± 359.98	197.4 ± 304.9	4075.2 ± 397.68	401.27 ± 236.8

	bpic2011_3		**bpic2011_2**		**sepsis_2**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	254.81 ± 402.07	1808.91 ± 0.78	1234.08 ± 371.71	2226.57 ± 1.21	26.68 ± 420.43	1020.36 ± 0.32
prefix_waveletLast	4694.83 ± 293.67	561.99 ± 189.27	10116.24 ± 192.47	411.27 ± 297.65	806.39 ± 219.13	**281.1 ± 386.26**
state_waveletAgg	80.15 ± 187.09	7549.01 ± 2.23	164.89 ± 196.23	7342.5 ± 2.3	**5.91 ± 199.78**	2068.21 ± 0.41
single_waveletLast	822.6 ± 256.97	3761.53 ± 1.03	5525.32 ± 383.11	2953.89 ± 1.02	79.44 ± 336.47	4056.21 ± 0.83
single_waveletAgg	2407.27 ± 420.65	10085.95 ± 3.41	1171.21 ± 387.25	4151.28 ± 2.32	30.46 ± 377.73	2708.93 ± 0.69
state_waveletLast	**34.48 ± 336.69**	1855.63 ± 0.44	**60.84 ± 337.31**	1899.34 ± 0.52	25.85 ± 225.44	5187.03 ± 0.87
prefix_waveletAgg	18678.3 ± 313.52	**192.79 ± 243.03**	38536.21 ± 341.3	**256.0 ± 231.34**	4248.4 ± 344.88	327.34 ± 390.2
cluster_WaveletAgg	1397427.78 ± 11344.92	21540.81 ± 20.01	7411613.76 ± 11749.54	19614.3 ± 20.22	276311.7 ± 14917.46	18380.38 ± 11.92
prefix_waveletIndex	1057.97 ± 318.78	434.44 ± 257.63	1185.13 ± 429.7	387.92 ± 398.16	923.03 ± 264.71	416.84 ± 370.45

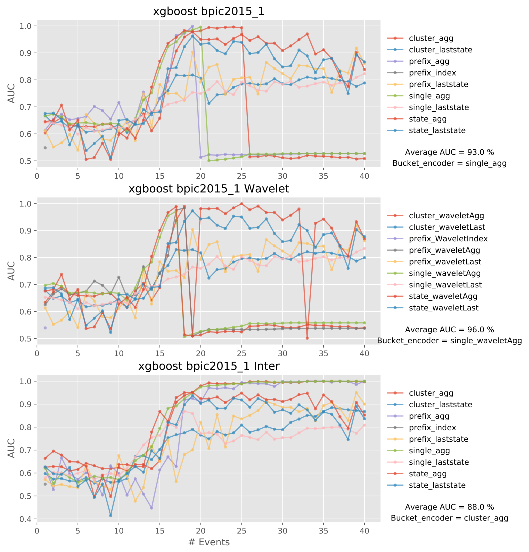
	bpic2015_1		**bpic2015_5**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	130.36 ± 230.65	645.75 ± 0.38	271.14 ± 222.07	678.18 ± 0.41	58.49 ± 284.24	857.96 ± 0.31
prefix_waveletLast	9885.91 ± 377.8	499.71 ± 257.42	15457.01 ± 380.89	314.33 ± 253.63	2398.25 ± 299.43	392.0 ± 386.65
state_waveletAgg	**20.21 ± 390.32**	732.17 ± 0.22	**24.81 ± 384.61**	740.47 ± 0.22	**9.5 ± 197.46**	1656.1 ± 0.38
single_waveletLast	861.59 ± 432.43	2960.77 ± 0.86	3171.43 ± 308.59	3466.53 ± 1.04	112.14 ± 220.83	3149.79 ± 0.69
single_waveletAgg	697.72 ± 283.39	1776.99 ± 0.67	6819.16 ± 419.08	1838.27 ± 0.71	120.13 ± 305.42	1652.73 ± 0.51
state_waveletLast	126.29 ± 265.64	2800.46 ± 0.77	97.26 ± 335.17	2051.32 ± 0.61	44.81 ± 302.71	4570.32 ± 0.91
prefix_waveletAgg	68667.33 ± 400.93	**253.5 ± 377.31**	117604.95 ± 405.81	**282.29 ± 261.42**	12802.85 ± 275.26	353.94 ± 212.67
cluster_WaveletAgg	1632441.22 ± 18367.52	18655.01 ± 20.26	3000711.32 ± 19865.65	13793.85 ± 14.63	635308.03 ± 10578.96	15517.33 ± 11.69
prefix_waveletIndex	2827.56 ± 322.61	385.59 ± 303.67	4219.53 ± 381.87	364.45 ± 319.57	961.01 ± 278.71	**249.88 ± 214.6**

	bpic2017_C		**production**		**sepsis_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	869.71 ± 440.09	375.76 ± 0.15	31.83 ± 403.23	1047.96 ± 0.3	68.09 ± 255.34	825.23 ± 0.3
prefix_waveletLast	24965.91 ± 387.77	482.13 ± 189.28	1719.25 ± 436.63	539.15 ± 414.86	2326.34 ± 394.15	468.42 ± 273.06
state_waveletAgg	**141.01 ± 231.53**	629.44 ± 0.14	**6.11 ± 400.39**	2404.65 ± 0.47	**8.64 ± 395.01**	1618.54 ± 0.36
single_waveletLast	83773.09 ± 255.06	668.1 ± 0.15	30.88 ± 328.56	1130.7 ± 0.17	920.74 ± 448.81	4749.2 ± 1.36
single_waveletAgg	15831.79 ± 241.24	537.59 ± 0.16	26.78 ± 219.23	992.28 ± 0.2	101.21 ± 295.16	1989.62 ± 0.59
state_waveletLast	206.09 ± 213.5	661.25 ± 0.13	9.76 ± 261.99	2210.08 ± 0.34	24.06 ± 368.89	2374.33 ± 0.57
prefix_waveletAgg	132253.65 ± 350.58	**322.52 ± 368.2**	4663.71 ± 265.65	286.06 ± 354.72	12512.19 ± 406.71	**168.37 ± 223.34**
cluster_WaveletAgg	274968066.52 ± 16996.27	9918.36 ± 8.19	587459.07 ± 12422.43	15705.24 ± 10.45	385330.49 ± 9182.82	15613.03 ± 12.14
prefix_waveletIndex	33083.92 ± 191.32	351.58 ± 329.41	211.88 ± 364.97	**202.0 ± 248.58**	930.06 ± 354.7	424.79 ± 382.09

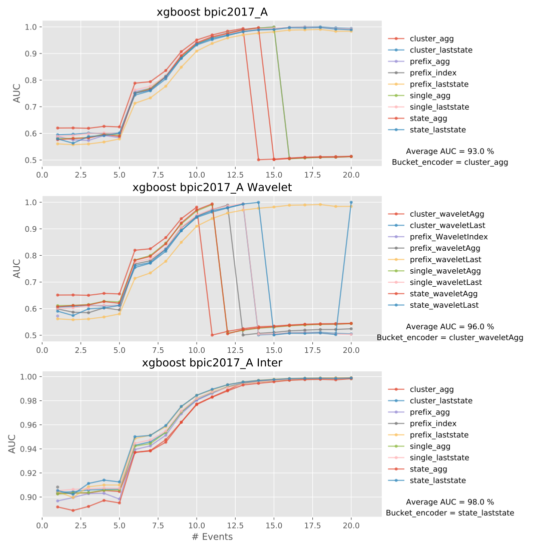
	bpic2011_1		**bpic2017_R**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_WaveletLast	715.56 ± 375.63	3079.01 ± 1.63	7752.35 ± 214.57	404.65 ± 0.16
prefix_waveletLast	8266.97 ± 261.83	510.56 ± 323.63	47470.91 ± 388.24	591.41 ± 246.56
state_waveletAgg	104.56 ± 266.27	7287.14 ± 2.15	4292.0 ± 184.71	722.0 ± 0.16
single_waveletLast	1839.19 ± 372.38	2073.35 ± 0.57	1547.55 ± 367.53	672.84 ± 0.15
single_waveletAgg	792.6 ± 249.75	1939.1 ± 0.69	3121.27 ± 312.16	697.88 ± 0.21
state_waveletLast	**56.21 ± 280.69**	2419.91 ± 0.61	**748.65 ± 378.67**	662.69 ± 0.13
prefix_waveletAgg	33997.08 ± 439.07	**201.94 ± 332.19**	143496.41 ± 423.53	**225.94 ± 369.06**
cluster_WaveletAgg	3203642.73 ± 12901.69	21336.04 ± 20.71	112106843.9 ± 19547.88	11448.56 ± 9.51
prefix_waveletIndex	1068.25 ± 340.09	251.48 ± 280.99	33550.74 ± 398.04	254.76 ± 252.98



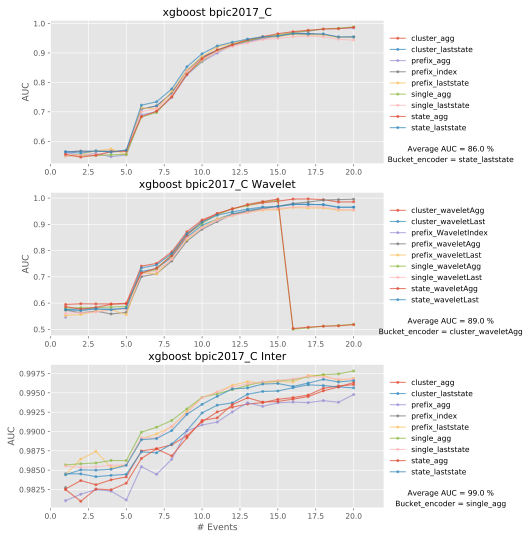
(a) Bpic2012_D



(b) Bpic2015

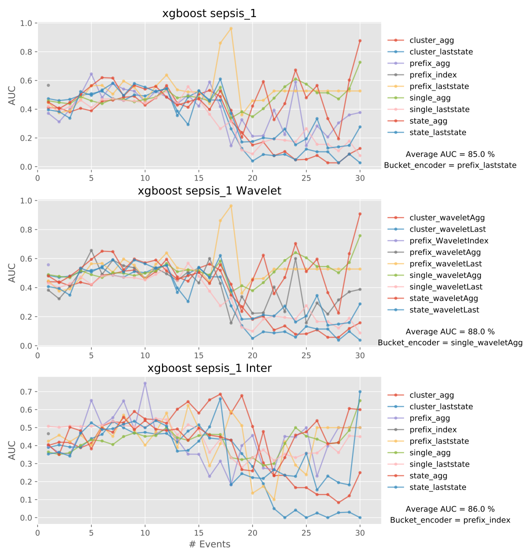


(c) Bpic2017_A

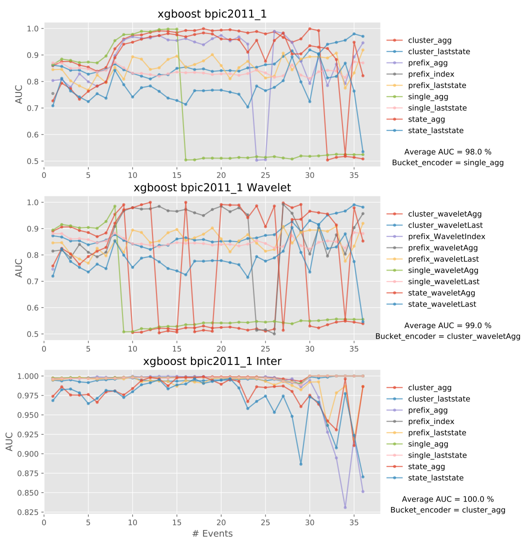


(d) Bpic2017_C

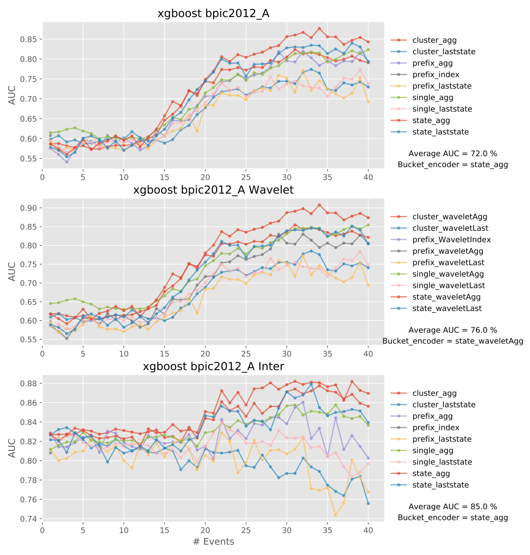
Figure B.25: Comparing XGBoost before and after adding Inter-case features on all event logs



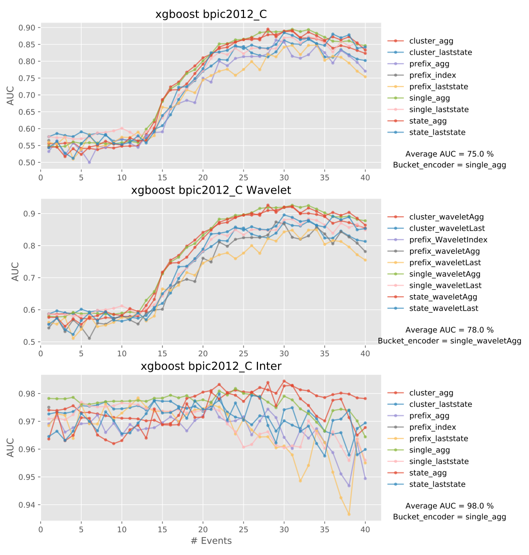
(a) Sepsis_1



(b) Bpic2011

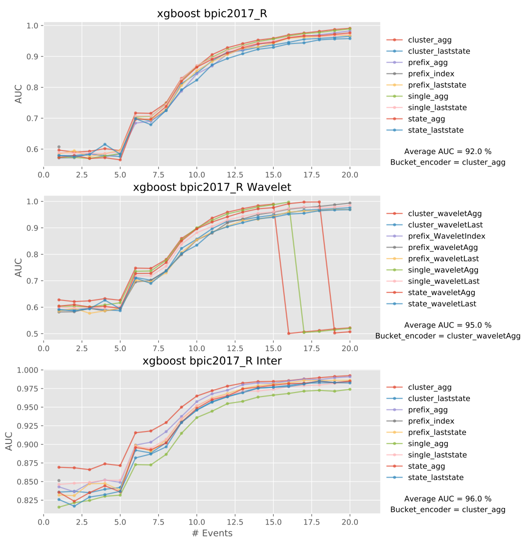


(c) Bpic2012_A

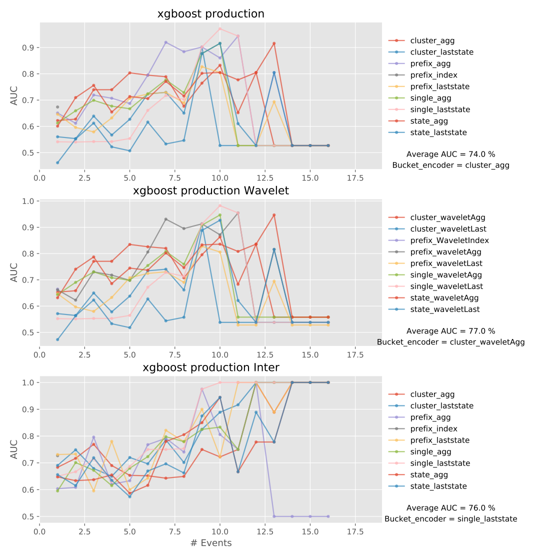


(d) Bpic2012_C

Figure B.26: Comparing XGBoost before and after adding Inter-case features on all event logs (*continued*).

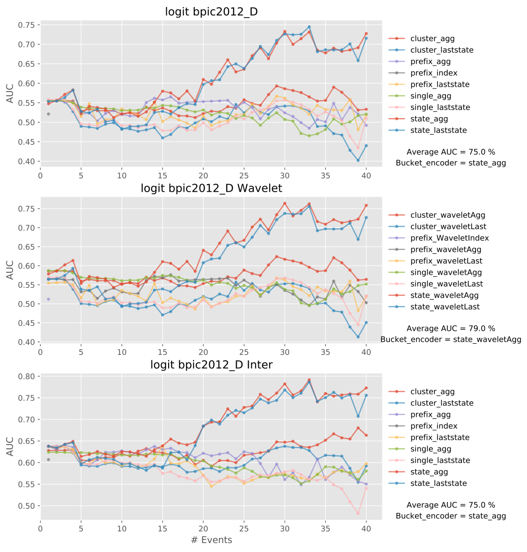


(a) Wavelet bpic2017_R

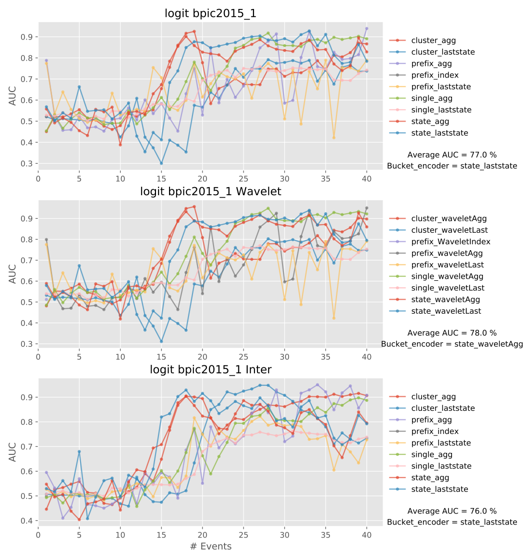


(b) Production

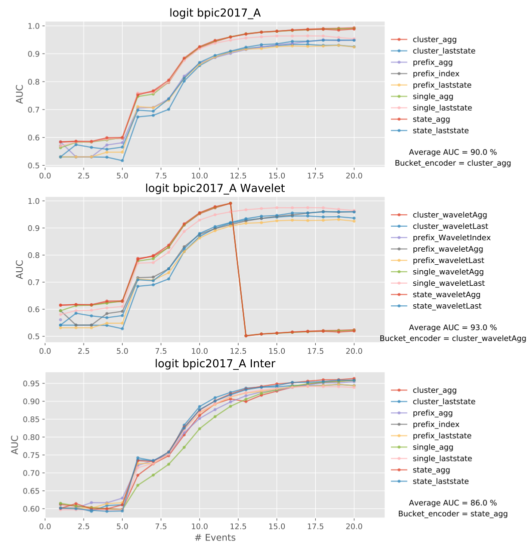
Figure B.27: Comparing XGBoost before and after adding Inter-case features on all event logs (*continued*)



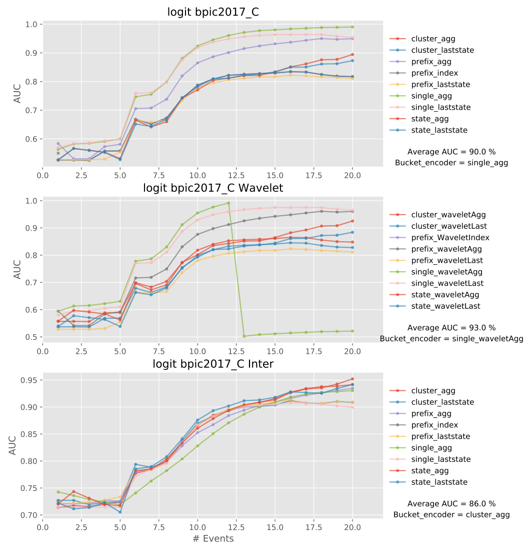
(a) Bpic2012_D



(b) Bpic2015

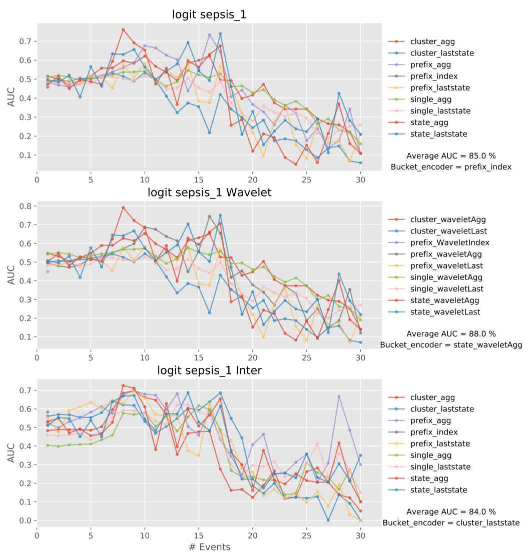


(c) Bpic2017_A

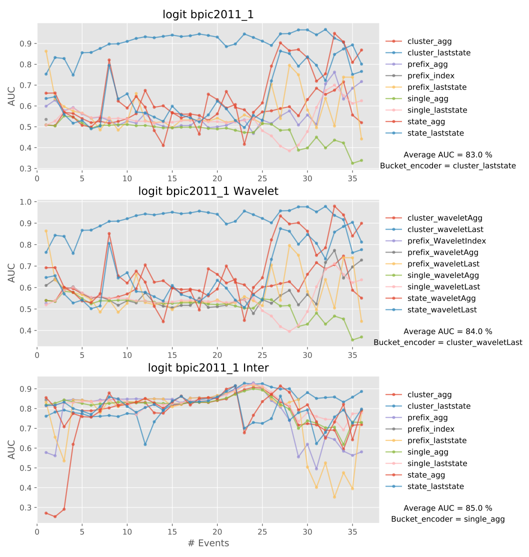


(d) Bpic2017_C

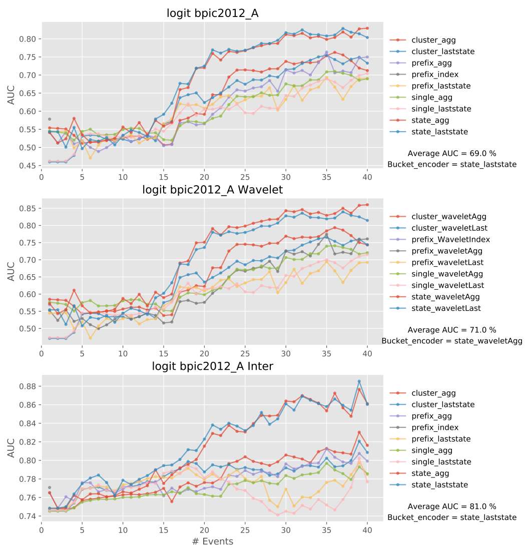
Figure B.28: Comparing Logit before and after adding Inter-case features on all event logs



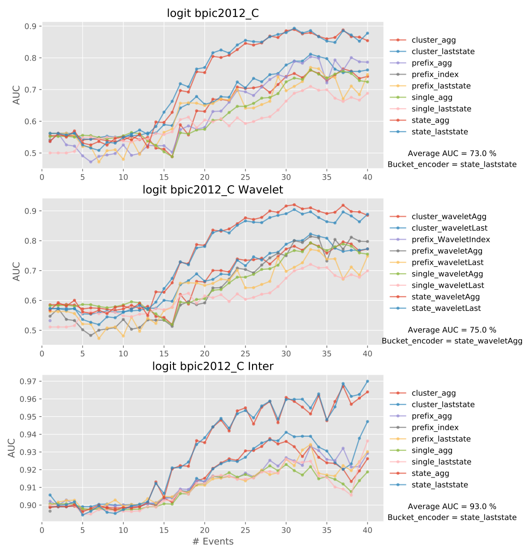
(a) Sepsis_1



(b) Bpic2011

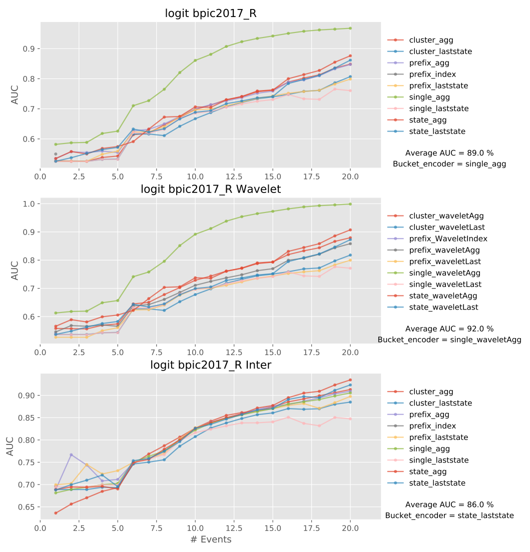


(c) Bpic2012_A

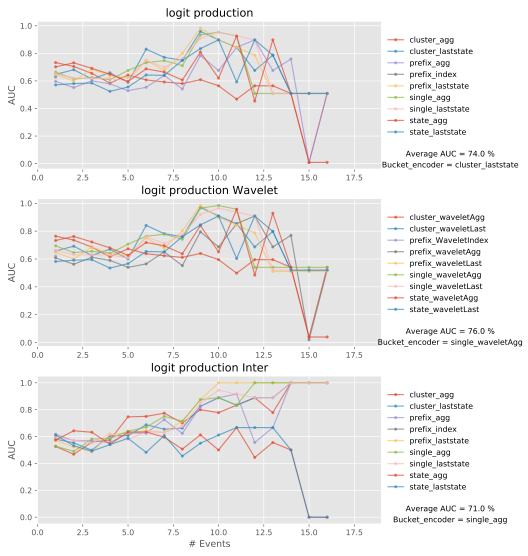


(d) Bpic2012_C

Figure B.29: Comparing Logit before and after adding Inter-case features on all event logs (*continued*).

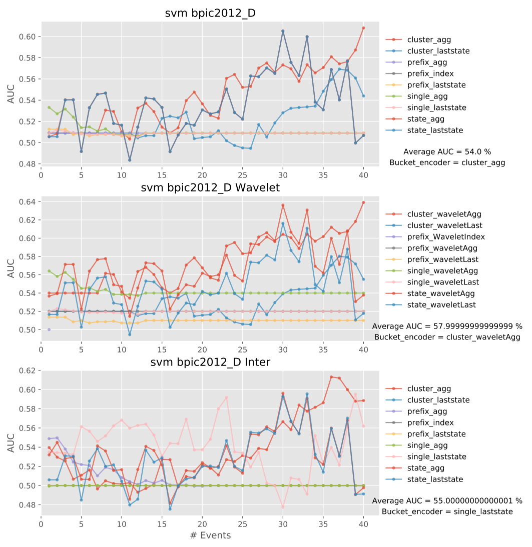


(a) Wavelet bpic2017_R

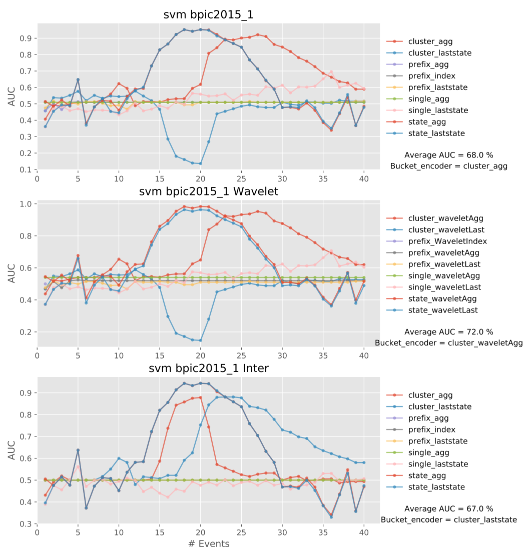


(b) Production

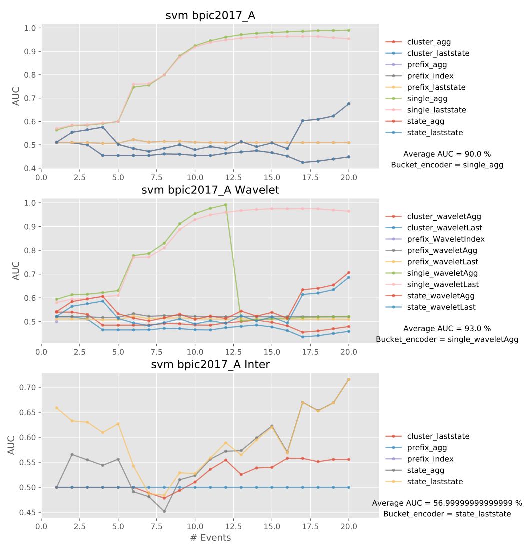
Figure B.30: Comparing Logit before and after adding Inter-case features on all event logs (*continued*)



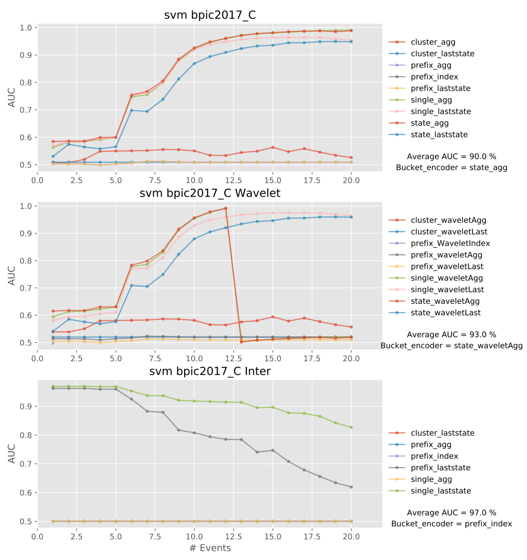
(a) Bpic2012_D



(b) Bpic2015

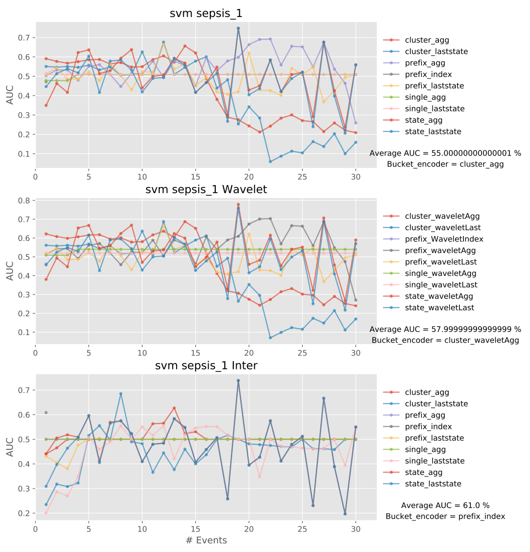


(c) Bpic2017_A

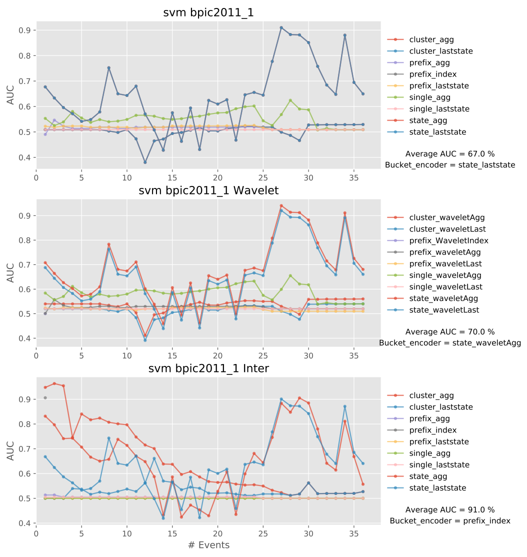


(d) Bpic2017_C

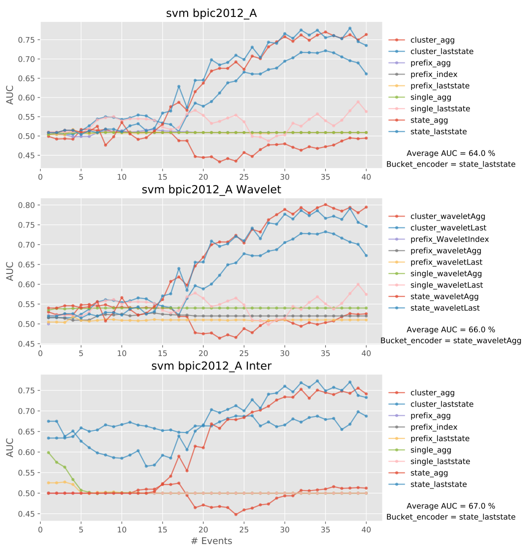
Figure B.31: Comparing SVM before and after adding Inter-case features on all event logs



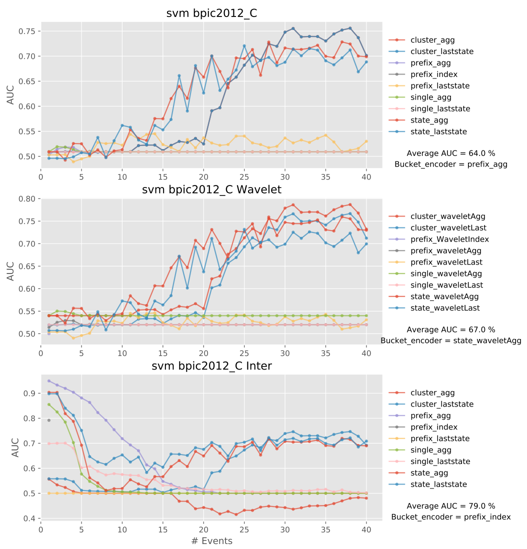
(a) Sepsis_1



(b) Bpic2011

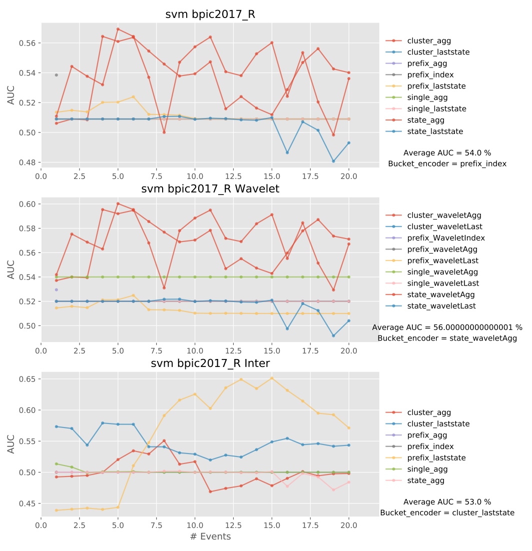


(c) Bpic2012_A

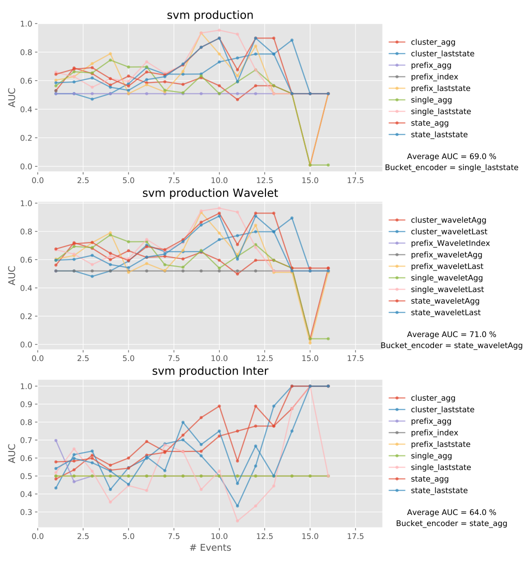


(d) Bpic2012_C

Figure B.32: Comparing SVM before and after adding Inter-case features on all event logs (*continued*).



(a) Wavelet bpic2017_R



(b) Production

Figure B.33: Comparing SVM before and after adding Inter-case features on all event logs (*continued*)

Table B.17: Overall AUC (F-score) for **XGBoost plus Inter-case features**

	xgboost					
	bpic2015_2	bpic2015_4	bpic2011_4	bpic2017_r	bpic2012_d	bpic2012_a
cluster_laststate	0.85 (0.46)	0.67 (0.31)	0.98 (0.9)	0.92 (0.66)	0.65 (0.23)	0.81 (0.78)
cluster_agg	0.83 (0.67)	0.65 (0.31)	0.96 (0.86)	**0.94** (**0.67**)	0.7 (0.22)	0.84 (0.82)
single_agg	**0.98** (**0.98**)	0.71 (0.27)	**0.99** (**0.92**)	0.91 (0.65)	0.67 (0.26)	0.83 (0.8)
prefix_agg	**0.98** (**0.98**)	0.67 (0.29)	0.98 (0.92)	0.93 (0.66)	0.66 (0.23)	0.82 (0.78)
prefix_laststate	0.85 (0.5)	0.71 (0.33)	**0.99** (**0.92**)	0.93 (0.66)	0.63 (0.23)	0.8 (0.76)
single_laststate	0.91 (0.67)	0.62 (0.28)	0.98 (0.92)	0.93 (0.66)	0.66 (0.24)	0.82 (0.8)
prefix_index	0.64 (0.2)	0.66 (0.36)	**0.99** (**0.92**)	0.85 (0.33)	0.67 (0.25)	0.81 (0.72)
state_agg	0.86 (0.34)	0.67 (0.26)	0.92 (0.81)	0.93 (0.67)	0.66 (0.29)	**0.85** (**0.82**)
state_laststate	0.86 (0.31)	0.67 (0.27)	0.91 (0.84)	0.92 (0.66)	0.68 (0.28)	0.83 (0.78)

	xgboost					
	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1	bpic2015_5
cluster_laststate	0.62 (0.32)	0.97 (0.9)	0.86 (0.86)	0.5 (0.0)	0.72 (0.32)	0.69 (0.38)
cluster_agg	0.61 (0.26)	0.98 (0.94)	0.98 (0.87)	0.5 (0.0)	0.83 (0.6)	0.68 (0.5)
single_agg	0.64 (0.28)	0.99 (0.97)	0.97 (0.85)	0.5 (0.0)	0.83 (0.66)	0.68 (0.46)
prefix_agg	0.65 (0.24)	0.97 (0.96)	0.97 (0.84)	0.5 (0.0)	0.79 (0.62)	0.7 (0.48)
prefix_laststate	0.64 (0.23)	0.98 (0.96)	0.87 (0.84)	0.5 (0.0)	0.72 (0.41)	0.7 (0.45)
single_laststate	0.63 (0.27)	**0.98** (**0.97**)	0.74 (0.77)	0.5 (0.0)	0.72 (0.38)	0.7 (0.44)
prefix_index	0.64 (0.28)	0.97 (0.96)	0.78 (0.83)	0.99 (0.85)	0.55 (0.24)	0.71 (0.46)
state_agg	0.64 (0.31)	0.93 (0.78)	0.65 (0.79)	0.33 (0.0)	0.8 (0.58)	0.64 (0.42)
state_laststate	0.65 (0.31)	0.95 (0.78)	0.51 (0.74)	0.33 (0.0)	0.75 (0.47)	0.68 (0.43)

	xgboost					
	sepsis_1	bpic2017_c	sepsis_3	bpic2012_c	production	bpic2011_1
cluster_laststate	0.42 (0.03)	**0.99** (**0.97**)	0.99 (0.88)	0.97 (0.87)	0.74 (0.64)	0.99 (0.93)
cluster_agg	0.49 (0.1)	**0.99** (**0.96**)	0.98 (0.88)	**0.98** (**0.88**)	0.71 (0.62)	**0.98** (**0.91**)
single_agg	0.42 (0.05)	**0.99** (**0.97**)	0.99 (0.92)	**0.98** (**0.89**)	0.71 (0.59)	**0.98** (**0.93**)
prefix_agg	0.47 (0.1)	**0.99** (**0.96**)	0.99 (0.92)	0.97 (0.87)	0.69 (0.63)	0.99 (0.92)
prefix_laststate	0.46 (0.08)	**0.99** (**0.97**)	0.99 (0.92)	0.97 (0.87)	0.74 (0.64)	0.99 (0.93)
single_laststate	0.49 (0.0)	**0.99** (**0.97**)	**0.98** (**0.92**)	0.97 (0.87)	0.74 (0.63)	**0.98** (**0.94**)
prefix_index	0.47 (0.19)	0.98 (0.95)	**0.98** (**0.92**)	**0.98** (**0.9**)	0.73 (0.69)	**0.98** (**0.94**)
state_agg	0.46 (0.08)	**0.99** (**0.96**)	0.94 (0.88)	0.97 (0.88)	0.69 (0.6)	0.98 (0.91)
state_laststate	0.44 (0.08)	**0.99** (**0.97**)	0.94 (0.88)	0.97 (0.88)	0.69 (0.62)	0.97 (0.87)

	xgboost
	bpic2017_a
cluster_laststate	0.96 (0.89)
cluster_agg	0.96 (0.88)
single_agg	0.96 (0.89)
prefix_agg	0.96 (0.88)
prefix_laststate	**0.97** (**0.9**)
single_laststate	0.96 (0.89)
prefix_index	0.91 (0.8)
state_agg	0.96 (0.88)
state_laststate	**0.97** (**0.9**)

Table B.18: Overall AUC (F-score) for **RF plus Inter-case features**

	rf					
	bpic2015_2	bpic2015_4	bpic2011_4	bpic2017_r	bpic2012_d	bpic2012_a
cluster_laststate	0.87 (0.4)	0.7 (0.3)	0.98 (0.87)	0.93 (0.66)	0.68 (0.27)	0.81 (0.79)
cluster_agg	**0.98** (**0.95**)	0.74 (0.27)	0.98 (0.85)	0.93 (0.65)	0.69 (0.28)	**0.85** (**0.82**)
single_agg	**0.98** (**1.0**)	0.75 (0.32)	**0.99** (**0.94**)	0.93 (0.66)	0.69 (0.27)	0.81 (0.77)
prefix_agg	**0.98** (**0.98**)	0.73 (0.32)	**0.99** (**0.93**)	**0.94** (**0.64**)	0.68 (0.27)	0.83 (0.8)
prefix_laststate	0.92 (0.24)	0.73 (0.43)	0.98 (0.93)	**0.94** (**0.64**)	0.68 (0.23)	0.84 (0.82)
single_laststate	0.92 (0.24)	0.71 (0.22)	0.94 (0.87)	**0.94** (**0.66**)	0.69 (0.25)	0.81 (0.8)
prefix_index	0.74 (0.49)	0.72 (0.33)	0.98 (0.92)	0.86 (0.34)	0.68 (0.23)	0.84 (0.79)
state_agg	0.85 (0.4)	0.67 (0.0)	0.94 (0.8)	**0.94** (**0.64**)	**0.71** (**0.21**)	0.83 (0.81)
state_laststate	0.81 (0.39)	0.72 (0.4)	0.93 (0.79)	**0.94** (**0.64**)	**0.71** (**0.18**)	**0.85** (**0.81**)

	rf					
	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1	bpic2015_5
cluster_laststate	0.64 (0.28)	**0.98** (**0.96**)	0.91 (0.87)	0.5 (0.0)	0.77 (0.43)	0.7 (0.43)
cluster_agg	0.65 (0.08)	0.99 (0.96)	**0.98** (**0.94**)	0.5 (0.0)	**0.87** (**0.67**)	0.67 (0.42)
single_agg	0.64 (0.28)	0.98 (0.97)	**0.98** (**1.0**)	0.5 (0.0)	0.86 (0.64)	0.69 (0.44)
prefix_agg	0.65 (0.13)	0.98 (0.97)	**0.98** (**0.88**)	0.5 (0.0)	0.81 (0.63)	0.69 (0.46)
prefix_laststate	0.63 (0.28)	**0.98** (**0.97**)	0.91 (0.88)	0.5 (0.0)	0.76 (0.4)	0.69 (0.46)
single_laststate	0.63 (0.26)	**0.98** (**0.97**)	0.94 (0.89)	0.5 (0.0)	0.77 (0.49)	0.68 (0.48)
prefix_index	0.63 (0.22)	**0.98** (**0.95**)	0.91 (0.85)	0.99 (0.93)	0.6 (0.16)	0.69 (0.46)
state_agg	**0.66** (**0.11**)	0.97 (0.89)	0.7 (0.8)	0.33 (0.0)	0.78 (0.6)	0.67 (0.47)
state_laststate	0.65 (0.28)	0.97 (0.87)	0.64 (0.73)	0.33 (0.0)	0.76 (0.51)	0.71 (0.43)

	rf					
	sepsis_1	bpic2017_c	sepsis_3	bpic2012_c	production	bpic2011_1
cluster_laststate	0.39 (0.0)	**0.99** (**0.97**)	0.96 (0.88)	0.97 (0.87)	0.71 (0.6)	0.99 (0.91)
cluster_agg	0.41 (0.0)	**0.99** (**0.97**)	0.98 (0.98)	0.97 (0.86)	0.72 (0.61)	**0.98** (**0.92**)
single_agg	0.39 (0.0)	**0.99** (**0.97**)	**0.98** (**1.0**)	**0.98** (**0.89**)	0.69 (0.61)	**0.98** (**0.93**)
prefix_agg	0.47 (0.01)	**0.99** (**0.97**)	**0.98** (**0.86**)	0.97 (0.85)	0.67 (0.62)	**0.98** (**0.97**)
prefix_laststate	0.47 (0.01)	**0.99** (**0.97**)	**0.98** (**0.92**)	**0.98** (**0.87**)	0.74 (0.61)	**0.98** (**0.93**)
single_laststate	0.49 (0.01)	**0.99** (**0.97**)	**0.98** (**1.0**)	**0.98** (**0.88**)	0.65 (0.61)	0.99 (0.92)
prefix_index	0.51 (0.0)	**0.99** (**0.96**)	**0.98** (**0.92**)	0.97 (0.89)	**0.78** (**0.67**)	**0.98** (**0.9**)
state_agg	0.43 (0.0)	**0.99** (**0.97**)	0.97 (0.98)	0.97 (0.87)	0.7 (0.64)	0.98 (0.93)
state_laststate	0.45 (0.0)	**0.99** (**0.97**)	0.98 (0.98)	**0.98** (**0.89**)	0.73 (0.59)	0.98 (0.91)

	rf
	bpic2017_a
cluster_laststate	**0.97** (**0.91**)
cluster_agg	**0.97** (**0.91**)
single_agg	**0.97** (**0.91**)
prefix_agg	**0.97** (**0.91**)
prefix_laststate	**0.97** (**0.91**)
single_laststate	**0.97** (**0.91**)
prefix_index	0.92 (0.84)
state_agg	**0.97** (**0.91**)
state_laststate	**0.97** (**0.91**)

Table B.19: Overall AUC (F-score) for **Logit plus Inter-case features**

	logit					
	bpic2015_2	bpic2015_4	bpic2011_4	bpic2017_r	bpic2012_d	bpic2012_a
cluster_laststate	0.69 (0.39)	0.7 (0.0)	0.72 (0.6)	0.79 (0.14)	0.6 (0.01)	0.78 (0.82)
cluster_agg	0.87 (0.29)	0.63 (0.0)	0.75 (0.64)	0.8 (0.14)	0.62 (0.0)	0.78 (0.81)
single_agg	0.85 (0.09)	0.63 (0.0)	0.72 (0.53)	0.81 (0.14)	0.6 (0.01)	0.77 (0.81)
prefix_agg	0.85 (0.09)	0.65 (0.0)	0.79 (0.72)	0.81 (0.2)	0.61 (0.0)	0.78 (0.82)
prefix_laststate	0.72 (0.0)	0.72 (0.0)	0.88 (0.77)	0.81 (0.18)	0.59 (0.0)	0.77 (0.82)
single_laststate	0.74 (0.0)	0.71 (0.0)	0.88 (0.77)	0.79 (0.03)	0.58 (0.01)	0.77 (0.81)
prefix_index	0.69 (0.0)	0.7 (0.0)	0.88 (0.78)	0.7 (0.02)	0.61 (0.0)	0.77 (0.8)
state_agg	0.86 (0.41)	0.66 (0.0)	0.73 (0.7)	0.81 (0.2)	0.68 (0.1)	0.8 (0.82)
state_laststate	0.87 (0.38)	0.73 (0.0)	0.76 (0.74)	0.81 (0.25)	0.66 (0.09)	0.81 (0.82)

	logit					
	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1	bpic2015_5
cluster_laststate	0.62 (0.2)	0.69 (0.0)	0.65 (0.76)	0.5 (0.0)	0.67 (0.31)	0.68 (0.33)
cluster_agg	0.6 (0.03)	0.7 (0.59)	0.64 (0.77)	0.5 (0.0)	0.72 (0.12)	0.56 (0.21)
single_agg	0.48 (0.0)	0.73 (0.02)	0.65 (0.77)	0.5 (0.0)	0.66 (0.12)	0.63 (0.34)
prefix_agg	0.49 (0.0)	0.67 (0.0)	0.65 (0.77)	0.5 (0.0)	0.67 (0.2)	0.64 (0.32)
prefix_laststate	0.54 (0.0)	0.7 (0.13)	0.58 (0.77)	0.5 (0.0)	0.64 (0.23)	0.6 (0.28)
single_laststate	0.54 (0.0)	0.77 (0.0)	0.68 (0.77)	0.5 (0.0)	0.63 (0.19)	0.56 (0.24)
prefix_index	0.54 (0.03)	0.92 (0.81)	0.81 (0.58)	0.95 (0.75)	0.5 (0.06)	0.58 (0.32)
state_agg	0.58 (0.03)	0.77 (0.58)	0.54 (0.71)	0.33 (0.0)	0.69 (0.4)	0.65 (0.31)
state_laststate	0.62 (0.2)	0.83 (0.36)	0.54 (0.68)	0.33 (0.0)	0.72 (0.35)	0.63 (0.38)

	logit					
	sepsis_1	bpic2017_c	sepsis_3	bpic2012_c	production	bpic2011_1
cluster_laststate	0.53 (0.06)	0.84 (0.77)	0.97 (0.9)	0.91 (0.8)	0.64 (0.6)	0.82 (0.7)
cluster_agg	0.5 (0.01)	0.84 (0.78)	0.94 (0.9)	0.91 (0.8)	0.64 (0.54)	0.74 (0.63)
single_agg	0.46 (0.0)	0.83 (0.75)	0.98 (0.92)	0.91 (0.79)	0.65 (0.59)	0.82 (0.69)
prefix_agg	0.56 (0.04)	0.84 (0.74)	0.98 (0.9)	0.91 (0.79)	0.64 (0.59)	0.8 (0.59)
prefix_laststate	0.54 (0.04)	0.84 (0.72)	0.98 (0.92)	0.91 (0.79)	0.65 (0.6)	0.79 (0.62)
single_laststate	0.49 (0.0)	0.83 (0.78)	0.98 (0.9)	0.91 (0.79)	0.65 (0.62)	0.83 (0.7)
prefix_index	0.58 (0.0)	0.73 (0.44)	0.98 (0.92)	0.9 (0.86)	0.59 (0.64)	0.84 (0.73)
state_agg	0.51 (0.03)	0.84 (0.76)	0.93 (0.88)	0.92 (0.82)	0.58 (0.52)	0.81 (0.67)
state_laststate	0.52 (0.04)	0.84 (0.73)	0.94 (0.9)	0.93 (0.82)	0.54 (0.51)	0.81 (0.66)

	logit
	bpic2017_a
cluster_laststate	0.82 (0.69)
cluster_agg	0.81 (0.68)
single_agg	0.8 (0.67)
prefix_agg	0.81 (0.69)
prefix_laststate	0.81 (0.7)
single_laststate	0.81 (0.68)
prefix_index	0.61 (0.52)
state_agg	0.82 (0.7)
state_laststate	0.82 (0.72)

Table B.20: Overall AUC (F-score) for **SVM plus Inter-case features**

	svm					
	bpic2015_2	bpic2015_4	bpic2011_4	bpic2017_r	bpic2012_d	bpic2012_a
cluster_laststate	0.56 (0.0)	0.58 (0.0)	0.51 (0.0)	0.55 (0.0)	0.5 (0.0)	0.66 (0.32)
cluster_agg	0.67 (0.46)	0.56 (0.0)	0.73 (0.07)	0.5 (0.0)	0.53 (0.0)	0.49 (0.0)
single_agg	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)	0.51 (0.71)
prefix_agg	0.5 (0.0)	0.5 (0.0)	0.88 (0.0)	0.5 (0.0)	0.51 (0.01)	0.5 (0.3)
prefix_laststate	0.5 (0.0)	0.5 (0.0)	0.84 (0.0)	0.56 (0.0)	0.5 (0.0)	0.5 (0.0)
single_laststate	0.5 (0.0)	0.53 (0.0)	0.84 (0.44)	–	0.54 (0.08)	0.5 (0.0)
prefix_index	0.68 (0.43)	0.53 (0.0)	0.88 (0.71)	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)
state_agg	0.7 (0.3)	0.53 (0.0)	0.82 (0.64)	0.5 (0.0)	0.53 (0.1)	0.6 (0.36)
state_laststate	0.71 (0.3)	0.53 (0.0)	0.49 (0.09)	–	0.52 (0.09)	0.66 (0.39)

	svm					
	bpic2015_3	bpic2011_3	bpic2011_2	sepsis_2	bpic2015_1	bpic2015_5
cluster_laststate	0.49 (0.0)	0.91 (0.59)	0.52 (0.2)	0.5 (0.0)	0.63 (0.0)	0.5 (0.0)
cluster_agg	0.52 (0.0)	0.52 (0.0)	0.56 (0.11)	0.5 (0.0)	0.56 (0.0)	0.5 (0.0)
single_agg	0.5 (0.0)	0.5 (0.0)	0.5 (0.77)	0.5 (0.0)	0.5 (0.0)	0.57 (0.0)
prefix_agg	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)
prefix_laststate	0.56 (0.0)	0.5 (0.0)	0.5 (0.77)	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)
single_laststate	0.5 (0.0)	0.5 (0.0)	0.5 (0.77)	0.5 (0.0)	0.48 (0.0)	0.57 (0.19)
prefix_index	0.53 (0.0)	0.5 (0.0)	0.5 (0.8)	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)
state_agg	0.56 (0.15)	0.72 (0.05)	0.51 (0.71)	0.33 (0.0)	0.62 (0.32)	0.57 (0.0)
state_laststate	0.49 (0.0)	0.58 (0.0)	0.55 (0.63)	0.33 (0.0)	0.62 (0.34)	0.5 (0.0)

	svm					
	sepsis_1	bpic2017_c	sepsis_3	bpic2012_c	production	bpic2011_1
cluster_laststate	0.44 (0.0)	0.5 (0.0)	0.52 (0.0)	0.59 (0.0)	0.58 (0.53)	0.53 (0.54)
cluster_agg	0.51 (0.0)	–	0.97 (0.34)	0.48 (0.0)	0.61 (0.52)	0.68 (0.54)
single_agg	0.5 (0.0)	0.5 (0.0)	0.52 (0.0)	0.54 (0.0)	0.5 (0.53)	0.5 (0.0)
prefix_agg	0.5 (0.0)	0.5 (0.0)	0.5 (0.0)	0.63 (0.0)	0.52 (0.0)	0.5 (0.0)
prefix_laststate	0.48 (0.0)	0.81 (0.06)	0.99 (0.4)	0.5 (0.0)	0.5 (0.53)	0.5 (0.0)
single_laststate	0.45 (0.0)	0.92 (0.34)	0.84 (0.0)	0.56 (0.0)	0.52 (0.06)	0.5 (0.0)
prefix_index	**0.61** (**0.0**)	0.97 (0.0)	0.5 (0.0)	0.79 (0.0)	0.5 (0.0)	0.91 (0.0)
state_agg	0.5 (0.0)	–	0.99 (0.0)	0.66 (0.08)	0.63 (0.42)	0.65 (0.66)
state_laststate	0.48 (0.0)	–	0.52 (0.0)	0.69 (0.13)	0.59 (0.54)	0.62 (0.05)

	svm
	bpic2017_a
cluster_laststate	0.52 (0.0)
cluster_agg	–
single_agg	–
prefix_agg	0.5 (0.0)
prefix_laststate	–
single_laststate	–
prefix_index	0.5 (0.0)
state_agg	0.57 (0.11)
state_laststate	0.6 (0.04)

Table B.21: Execution times for **inter-case features plus XGBoost**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	4.88 ± 0.0	0.01 ± 0.01	60.94 ± 0.0	0.01 ± 0.01	3.64 ± 0.0	0.01 ± 0.01
cluster_agg	5.41 ± 0.0	0.01 ± 0.02	75.17 ± 0.0	0.01 ± 0.01	4.77 ± 0.0	0.01 ± 0.02
single_agg	18.77 ± 0.0	0.01 ± 0.01	106.34 ± 0.0	$\mathbf{0.0 \pm 0.01}$	7.03 ± 0.0	0.01 ± 0.01
prefix_agg	0.85 ± 0.0	$\mathbf{0.0 \pm 0.0}$	2.5 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.74 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	0.86 ± 0.0	$\mathbf{0.0 \pm 0.0}$	2.55 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.63 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	20.12 ± 0.0	0.01 ± 0.01	46.08 ± 0.0	$\mathbf{0.0 \pm 0.01}$	9.1 ± 0.0	0.01 ± 0.01
prefix_index	$\mathbf{0.12 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.72 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.09 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	12.84 ± 0.0	0.01 ± 0.02	24.22 ± 0.0	0.01 ± 0.01	9.86 ± 0.0	0.01 ± 0.02
state_laststate	8.65 ± 0.0	0.01 ± 0.02	16.13 ± 0.0	$\mathbf{0.0 \pm 0.01}$	4.5 ± 0.0	0.01 ± 0.02

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	67.26 ± 0.0	0.01 ± 0.01	832.53 ± 0.0	0.01 ± 0.01	91.92 ± 0.0	0.02 ± 0.03
cluster_agg	74.64 ± 0.0	0.01 ± 0.01	844.27 ± 0.0	0.01 ± 0.01	31.47 ± 0.0	0.02 ± 0.03
single_agg	54.8 ± 0.0	$\mathbf{0.0 \pm 0.01}$	1468.06 ± 0.0	0.01 ± 0.01	26.69 ± 0.0	0.02 ± 0.03
prefix_agg	2.6 ± 0.0	$\mathbf{0.0 \pm 0.0}$	6.78 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.15 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	2.47 ± 0.0	$\mathbf{0.0 \pm 0.0}$	6.81 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.13 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	162.49 ± 0.0	$\mathbf{0.0 \pm 0.01}$	1085.55 ± 0.0	0.01 ± 0.01	22.01 ± 0.0	0.02 ± 0.03
prefix_index	$\mathbf{0.71 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{5.25 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.19 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	55.03 ± 0.0	0.01 ± 0.01	551.18 ± 0.0	0.01 ± 0.01	17.44 ± 0.0	0.02 ± 0.03
state_laststate	35.62 ± 0.0	$\mathbf{0.0 \pm 0.01}$	270.3 ± 0.0	0.01 ± 0.01	26.1 ± 0.0	0.02 ± 0.03

	bpic2012_A		**bpic2015_3**		**bpic2011_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	44.37 ± 0.0	0.01 ± 0.01	19.02 ± 0.0	0.01 ± 0.02	8.66 ± 0.0	0.02 ± 0.03
cluster_agg	91.93 ± 0.0	0.01 ± 0.01	14.59 ± 0.0	0.01 ± 0.02	10.93 ± 0.0	0.02 ± 0.03
single_agg	71.59 ± 0.0	$\mathbf{0.0 \pm 0.01}$	105.18 ± 0.0	0.01 ± 0.01	22.17 ± 0.0	0.02 ± 0.03
prefix_agg	2.44 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.58 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.59 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	2.45 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.37 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.58 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	140.47 ± 0.0	$\mathbf{0.0 \pm 0.01}$	64.33 ± 0.0	0.01 ± 0.01	18.13 ± 0.0	0.02 ± 0.03
prefix_index	$\mathbf{0.73 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.21 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.18 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	75.04 ± 0.0	0.01 ± 0.01	19.99 ± 0.0	0.01 ± 0.02	16.43 ± 0.0	0.02 ± 0.03
state_laststate	34.25 ± 0.0	$\mathbf{0.0 \pm 0.01}$	10.85 ± 0.0	0.01 ± 0.02	9.51 ± 0.0	0.02 ± 0.03

	sepsis_2		**bpic2011_2**		**bpic2017_C**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	1.62 ± 0.0	0.02 ± 0.02	27.91 ± 0.0	0.02 ± 0.03	216.15 ± 0.0	0.01 ± 0.01
cluster_agg	2.38 ± 0.0	0.02 ± 0.02	17.26 ± 0.0	0.02 ± 0.03	446.0 ± 0.0	0.01 ± 0.01
single_agg	1.72 ± 0.0	0.02 ± 0.02	31.26 ± 0.0	0.02 ± 0.03	914.57 ± 0.0	0.01 ± 0.01
prefix_agg	0.16 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.23 ± 0.0	$\mathbf{0.0 \pm 0.0}$	6.78 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	0.16 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.2 ± 0.0	$\mathbf{0.0 \pm 0.0}$	6.83 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	1.52 ± 0.0	0.02 ± 0.02	42.13 ± 0.0	0.02 ± 0.03	779.63 ± 0.0	0.01 ± 0.01
prefix_index	$\mathbf{0.14 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.2 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{5.4 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	2.34 ± 0.0	0.02 ± 0.02	18.86 ± 0.0	0.02 ± 0.04	289.58 ± 0.0	0.01 ± 0.01
state_laststate	1.52 ± 0.0	0.02 ± 0.02	24.0 ± 0.0	0.02 ± 0.03	280.38 ± 0.0	0.01 ± 0.01

	bpic2015_5		**sepsis_1**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	28.22 ± 0.0	0.01 ± 0.02	3.58 ± 0.0	0.01 ± 0.02	2.11 ± 0.0	0.02 ± 0.02
cluster_agg	11.2 ± 0.0	0.01 ± 0.02	3.1 ± 0.0	0.02 ± 0.02	2.62 ± 0.0	0.02 ± 0.02
single_agg	29.88 ± 0.0	0.01 ± 0.02	3.08 ± 0.0	0.01 ± 0.02	2.72 ± 0.0	0.01 ± 0.02
prefix_agg	1.3 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.47 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.51 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	1.14 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.47 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.51 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	30.39 ± 0.0	0.01 ± 0.01	4.99 ± 0.0	0.01 ± 0.01	2.69 ± 0.0	0.01 ± 0.02
prefix_index	$\mathbf{0.17 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.13 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.13 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	11.47 ± 0.0	0.01 ± 0.02	4.58 ± 0.0	0.02 ± 0.02	3.93 ± 0.0	0.02 ± 0.02
state_laststate	8.13 ± 0.0	0.01 ± 0.02	2.73 ± 0.0	0.01 ± 0.02	2.29 ± 0.0	0.01 ± 0.02

	bpic2015_1		**production**		**bpic2011_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	4.71 ± 0.0	0.01 ± 0.02	0.67 ± 0.0	0.01 ± 0.01	16.29 ± 0.0	0.02 ± 0.04
cluster_agg	13.85 ± 0.0	0.01 ± 0.02	0.55 ± 0.0	0.02 ± 0.01	10.94 ± 0.0	0.02 ± 0.04
single_agg	14.5 ± 0.0	0.01 ± 0.02	0.55 ± 0.0	0.01 ± 0.01	17.88 ± 0.0	0.02 ± 0.03
prefix_agg	0.76 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.12 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.82 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	0.76 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.12 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.81 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	6.36 ± 0.0	0.01 ± 0.01	0.38 ± 0.0	0.01 ± 0.01	20.77 ± 0.0	0.02 ± 0.03
prefix_index	$\mathbf{0.11 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.04 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.18 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	11.19 ± 0.0	0.01 ± 0.02	0.72 ± 0.0	0.01 ± 0.01	15.51 ± 0.0	0.02 ± 0.03
state_laststate	8.72 ± 0.0	0.01 ± 0.02	0.54 ± 0.0	0.01 ± 0.01	17.31 ± 0.0	0.02 ± 0.03

	bpic2017_R	
method	training_total (s)	prediction_avg (ms)
cluster_laststate	377.35 ± 0.0	0.01 ± 0.01
cluster_agg	986.18 ± 0.0	0.01 ± 0.01
single_agg	930.87 ± 0.0	0.01 ± 0.01
prefix_agg	7.54 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	6.83 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	1245.79 ± 0.0	0.01 ± 0.01
prefix_index	$\mathbf{5.33 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	514.08 ± 0.0	0.01 ± 0.01
state_laststate	363.26 ± 0.0	0.01 ± 0.01

Table B.22: Execution times for **inter-case features plus Logit**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	2.69 ± 0.0	0.01 ± 0.01	5.85 ± 0.0	0.01 ± 0.01	3.06 ± 0.0	0.01 ± 0.01
cluster_agg	2.68 ± 0.0	0.01 ± 0.02	8.52 ± 0.0	0.01 ± 0.01	2.0 ± 0.0	0.01 ± 0.01
single_agg	1.61 ± 0.0	0.01 ± 0.01	7.87 ± 0.0	**0.0 ± 0.01**	1.76 ± 0.0	0.01 ± 0.01
prefix_agg	0.87 ± 0.0	**0.0 ± 0.0**	2.54 ± 0.0	**0.0 ± 0.0**	0.63 ± 0.0	**0.0 ± 0.0**
prefix_laststate	0.86 ± 0.0	**0.0 ± 0.0**	2.52 ± 0.0	**0.0 ± 0.0**	0.63 ± 0.0	**0.0 ± 0.0**
single_laststate	1.47 ± 0.0	0.01 ± 0.01	3.54 ± 0.0	**0.0 ± 0.01**	2.31 ± 0.0	0.01 ± 0.01
prefix_index	**0.12 ± 0.0**	**0.0 ± 0.0**	**0.72 ± 0.0**	**0.0 ± 0.0**	**0.1 ± 0.0**	**0.0 ± 0.0**
state_agg	5.76 ± 0.0	0.01 ± 0.02	11.5 ± 0.0	0.01 ± 0.01	3.93 ± 0.0	0.01 ± 0.02
state_laststate	4.16 ± 0.0	0.01 ± 0.02	8.89 ± 0.0	**0.0 ± 0.01**	3.63 ± 0.0	0.01 ± 0.02

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	6.47 ± 0.0	0.01 ± 0.01	41.66 ± 0.0	0.01 ± 0.02	6.97 ± 0.0	0.02 ± 0.03
cluster_agg	8.42 ± 0.0	0.01 ± 0.01	35.47 ± 0.0	0.01 ± 0.01	7.35 ± 0.0	0.02 ± 0.03
single_agg	7.88 ± 0.0	**0.0 ± 0.01**	39.45 ± 0.0	0.01 ± 0.01	51.11 ± 0.0	0.02 ± 0.03
prefix_agg	2.5 ± 0.0	**0.0 ± 0.0**	7.68 ± 0.0	**0.0 ± 0.0**	1.14 ± 0.0	**0.0 ± 0.0**
prefix_laststate	2.42 ± 0.0	**0.0 ± 0.0**	7.47 ± 0.0	**0.0 ± 0.0**	1.16 ± 0.0	**0.0 ± 0.0**
single_laststate	5.56 ± 0.0	**0.0 ± 0.01**	29.0 ± 0.0	0.01 ± 0.01	7.55 ± 0.0	0.02 ± 0.03
prefix_index	**0.72 ± 0.0**	**0.0 ± 0.0**	**5.3 ± 0.0**	**0.0 ± 0.0**	**0.19 ± 0.0**	**0.0 ± 0.0**
state_agg	11.35 ± 0.0	0.01 ± 0.01	36.69 ± 0.0	0.01 ± 0.01	7.33 ± 0.0	0.02 ± 0.04
state_laststate	7.79 ± 0.0	**0.0 ± 0.01**	29.13 ± 0.0	0.01 ± 0.01	9.44 ± 0.0	0.02 ± 0.03

	bpic2012_A		**bpic2015_3**		**bpic2011_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	9.57 ± 0.0	**0.0 ± 0.01**	3.86 ± 0.0	0.01 ± 0.02	2.16 ± 0.0	0.03 ± 0.04
cluster_agg	7.04 ± 0.0	0.01 ± 0.01	4.7 ± 0.0	0.01 ± 0.02	2.35 ± 0.0	0.02 ± 0.03
single_agg	7.74 ± 0.0	**0.0 ± 0.01**	2.52 ± 0.0	0.01 ± 0.01	1.73 ± 0.0	0.02 ± 0.03
prefix_agg	2.47 ± 0.0	**0.0 ± 0.0**	1.38 ± 0.0	**0.0 ± 0.0**	0.59 ± 0.0	**0.0 ± 0.0**
prefix_laststate	2.45 ± 0.0	**0.0 ± 0.0**	1.38 ± 0.0	**0.0 ± 0.0**	0.6 ± 0.0	**0.0 ± 0.0**
single_laststate	5.95 ± 0.0	**0.0 ± 0.01**	2.66 ± 0.0	0.01 ± 0.01	42.79 ± 0.0	0.02 ± 0.03
prefix_index	**0.73 ± 0.0**	**0.0 ± 0.0**	**0.21 ± 0.0**	**0.0 ± 0.0**	**0.18 ± 0.0**	**0.0 ± 0.0**
state_agg	10.7 ± 0.0	0.01 ± 0.01	11.67 ± 0.0	0.01 ± 0.02	11.17 ± 0.0	0.02 ± 0.03
state_laststate	9.29 ± 0.0	**0.0 ± 0.01**	10.49 ± 0.0	0.01 ± 0.02	4.95 ± 0.0	0.02 ± 0.03

	sepsis_2		**bpic2011_2**		**bpic2017_C**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	2.31 ± 0.0	0.02 ± 0.02	55.83 ± 0.0	0.02 ± 0.03	27.44 ± 0.0	0.01 ± 0.01
cluster_agg	1.0 ± 0.0	0.02 ± 0.02	4.94 ± 0.0	0.02 ± 0.03	39.46 ± 0.0	0.01 ± 0.01
single_agg	0.5 ± 0.0	0.02 ± 0.02	8.59 ± 0.0	0.02 ± 0.03	39.67 ± 0.0	0.01 ± 0.01
prefix_agg	0.15 ± 0.0	**0.0 ± 0.0**	1.21 ± 0.0	**0.0 ± 0.0**	6.86 ± 0.0	**0.0 ± 0.0**
prefix_laststate	0.16 ± 0.0	**0.0 ± 0.0**	1.25 ± 0.0	**0.0 ± 0.0**	6.83 ± 0.0	**0.0 ± 0.0**
single_laststate	0.91 ± 0.0	0.02 ± 0.02	2.47 ± 0.0	0.02 ± 0.03	27.5 ± 0.0	0.01 ± 0.01
prefix_index	**0.12 ± 0.0**	**0.0 ± 0.0**	**0.2 ± 0.0**	**0.0 ± 0.0**	**5.29 ± 0.0**	**0.0 ± 0.0**
state_agg	4.33 ± 0.0	0.02 ± 0.02	26.16 ± 0.0	0.02 ± 0.04	39.27 ± 0.0	0.01 ± 0.01
state_laststate	3.53 ± 0.0	0.02 ± 0.02	14.45 ± 0.0	0.02 ± 0.03	29.29 ± 0.0	0.01 ± 0.01

	bpic2015_5		**sepsis_1**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	4.65 ± 0.0	0.01 ± 0.02	1.72 ± 0.0	0.01 ± 0.02	1.76 ± 0.0	0.02 ± 0.02
cluster_agg	3.34 ± 0.0	0.01 ± 0.02	2.7 ± 0.0	0.02 ± 0.02	1.76 ± 0.0	0.02 ± 0.02
single_agg	3.46 ± 0.0	0.01 ± 0.01	1.52 ± 0.0	0.01 ± 0.02	0.87 ± 0.0	0.01 ± 0.02
prefix_agg	1.12 ± 0.0	**0.0 ± 0.0**	0.47 ± 0.0	**0.0 ± 0.0**	0.5 ± 0.0	**0.0 ± 0.0**
prefix_laststate	1.12 ± 0.0	**0.0 ± 0.0**	0.47 ± 0.0	**0.0 ± 0.0**	0.51 ± 0.0	**0.0 ± 0.0**
single_laststate	1.58 ± 0.0	0.01 ± 0.01	1.16 ± 0.0	0.01 ± 0.01	1.25 ± 0.0	0.01 ± 0.02
prefix_index	**0.17 ± 0.0**	**0.0 ± 0.0**	**0.12 ± 0.0**	**0.0 ± 0.0**	**0.12 ± 0.0**	**0.0 ± 0.0**
state_agg	10.88 ± 0.0	0.01 ± 0.02	8.24 ± 0.0	0.02 ± 0.02	1.25 ± 0.0	0.02 ± 0.02
state_laststate	8.2 ± 0.0	0.01 ± 0.02	7.0 ± 0.0	0.01 ± 0.02	1.37 ± 0.0	0.01 ± 0.02

	bpic2015_1		**production**		**bpic2011_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	4.54 ± 0.0	0.01 ± 0.02	1.82 ± 0.0	0.01 ± 0.01	4.2 ± 0.0	0.02 ± 0.04
cluster_agg	3.42 ± 0.0	0.01 ± 0.02	2.29 ± 0.0	0.02 ± 0.01	5.74 ± 0.0	0.02 ± 0.03
single_agg	3.12 ± 0.0	0.01 ± 0.01	1.67 ± 0.0	0.01 ± 0.01	6.74 ± 0.0	0.02 ± 0.03
prefix_agg	0.74 ± 0.0	**0.0 ± 0.0**	0.12 ± 0.0	**0.0 ± 0.0**	0.8 ± 0.0	**0.0 ± 0.0**
prefix_laststate	0.75 ± 0.0	**0.0 ± 0.0**	0.12 ± 0.0	**0.0 ± 0.0**	0.83 ± 0.0	**0.0 ± 0.0**
single_laststate	2.39 ± 0.0	0.01 ± 0.01	1.61 ± 0.0	0.01 ± 0.01	6.24 ± 0.0	0.02 ± 0.03
prefix_index	**0.11 ± 0.0**	**0.0 ± 0.0**	**0.04 ± 0.0**	**0.0 ± 0.0**	**0.18 ± 0.0**	**0.0 ± 0.0**
state_agg	12.29 ± 0.0	0.01 ± 0.02	3.8 ± 0.0	0.01 ± 0.01	15.4 ± 0.0	0.02 ± 0.03
state_laststate	11.51 ± 0.0	0.01 ± 0.02	3.64 ± 0.0	0.01 ± 0.01	15.15 ± 0.0	0.02 ± 0.03

	bpic2017_R	
method	training_total (s)	prediction_avg (ms)
cluster_laststate	27.06 ± 0.0	0.01 ± 0.01
cluster_agg	41.08 ± 0.0	0.01 ± 0.01
single_agg	39.49 ± 0.0	0.01 ± 0.01
prefix_agg	6.72 ± 0.0	**0.0 ± 0.0**
prefix_laststate	6.81 ± 0.0	**0.0 ± 0.0**
single_laststate	21.98 ± 0.0	0.01 ± 0.01
prefix_index	**5.3 ± 0.0**	**0.0 ± 0.0**
state_agg	36.45 ± 0.0	0.01 ± 0.01
state_laststate	22.82 ± 0.0	0.01 ± 0.01

Table B.23: Execution times for **inter-case features plus RF**

	bpic2015_2		**bpic2012_C**		**bpic2015_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	3.28 ± 0.0	0.02 ± 0.04	7.07 ± 0.0	0.02 ± 0.03	4.43 ± 0.0	0.02 ± 0.04
cluster_agg	9.97 ± 0.0	0.03 ± 0.05	13.42 ± 0.0	0.02 ± 0.03	8.57 ± 0.0	0.03 ± 0.05
single_agg	4.01 ± 0.0	0.03 ± 0.05	5.41 ± 0.0	0.02 ± 0.03	2.7 ± 0.0	0.03 ± 0.04
prefix_agg	0.85 ± 0.0	$\mathbf{0.0 \pm 0.0}$	2.47 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.62 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	0.85 ± 0.0	$\mathbf{0.0 \pm 0.0}$	2.6 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.62 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	2.93 ± 0.0	0.03 ± 0.04	6.99 ± 0.0	0.02 ± 0.03	1.55 ± 0.0	0.02 ± 0.03
prefix_index	$\mathbf{0.12 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.76 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.09 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	63.13 ± 0.0	0.02 ± 0.04	25.0 ± 0.0	0.02 ± 0.03	53.47 ± 0.0	0.02 ± 0.04
state_laststate	76.08 ± 0.0	0.02 ± 0.04	43.53 ± 0.0	0.02 ± 0.03	57.87 ± 0.0	0.02 ± 0.04

	bpic2012_D		**bpic2017_A**		**bpic2011_4**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	19.33 ± 0.0	0.03 ± 0.05	25.7 ± 0.0	0.03 ± 0.04	5.4 ± 0.0	0.05 ± 0.08
cluster_agg	30.25 ± 0.0	0.03 ± 0.04	41.57 ± 0.0	0.03 ± 0.04	10.71 ± 0.0	0.04 ± 0.07
single_agg	36.68 ± 0.0	0.02 ± 0.04	26.14 ± 0.0	0.03 ± 0.04	4.19 ± 0.0	0.03 ± 0.06
prefix_agg	2.68 ± 0.0	$\mathbf{0.0 \pm 0.0}$	10.32 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.14 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	2.75 ± 0.0	$\mathbf{0.0 \pm 0.0}$	8.21 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.16 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	6.79 ± 0.0	0.02 ± 0.03	33.8 ± 0.0	0.04 ± 0.06	3.31 ± 0.0	0.04 ± 0.07
prefix_index	$\mathbf{0.82 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{5.41 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.19 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	42.41 ± 0.0	0.02 ± 0.03	66.3 ± 0.0	0.04 ± 0.06	58.48 ± 0.0	0.03 ± 0.05
state_laststate	23.73 ± 0.0	0.02 ± 0.03	55.35 ± 0.0	0.04 ± 0.06	61.66 ± 0.0	0.03 ± 0.05

	bpic2012_A		**bpic2015_3**		**bpic2011_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	17.41 ± 0.0	0.03 ± 0.05	12.81 ± 0.0	0.03 ± 0.04	8.56 ± 0.0	0.06 ± 0.09
cluster_agg	8.14 ± 0.0	0.02 ± 0.03	6.99 ± 0.0	0.02 ± 0.04	8.37 ± 0.0	0.06 ± 0.08
single_agg	31.32 ± 0.0	0.02 ± 0.03	7.36 ± 0.0	0.02 ± 0.03	2.8 ± 0.0	0.04 ± 0.06
prefix_agg	2.49 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.34 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.6 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	2.47 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.35 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.6 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	8.09 ± 0.0	0.02 ± 0.03	5.13 ± 0.0	0.02 ± 0.03	1.9 ± 0.0	0.04 ± 0.06
prefix_index	$\mathbf{0.71 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.21 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.18 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	51.73 ± 0.0	0.02 ± 0.03	127.3 ± 0.0	0.03 ± 0.04	61.4 ± 0.0	0.04 ± 0.06
state_laststate	23.18 ± 0.0	0.02 ± 0.03	129.68 ± 0.0	0.02 ± 0.04	64.39 ± 0.0	0.05 ± 0.08

	sepsis_2		**bpic2011_2**		**bpic2017_C**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	7.24 ± 0.0	0.05 ± 0.05	5.44 ± 0.0	0.04 ± 0.06	48.26 ± 0.0	0.03 ± 0.04
cluster_agg	3.42 ± 0.0	0.05 ± 0.05	7.41 ± 0.0	0.04 ± 0.06	29.49 ± 0.0	0.03 ± 0.04
single_agg	1.95 ± 0.0	0.05 ± 0.05	7.13 ± 0.0	0.03 ± 0.06	22.15 ± 0.0	0.03 ± 0.04
prefix_agg	0.16 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.23 ± 0.0	$\mathbf{0.0 \pm 0.0}$	7.47 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	0.16 ± 0.0	$\mathbf{0.0 \pm 0.0}$	1.24 ± 0.0	$\mathbf{0.0 \pm 0.0}$	6.83 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	1.87 ± 0.0	0.05 ± 0.05	2.83 ± 0.0	0.03 ± 0.05	23.67 ± 0.0	0.03 ± 0.04
prefix_index	$\mathbf{0.12 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.21 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{5.34 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	12.58 ± 0.0	0.05 ± 0.05	69.77 ± 0.0	0.04 ± 0.06	142.51 ± 0.0	0.03 ± 0.04
state_laststate	13.02 ± 0.0	0.05 ± 0.05	72.54 ± 0.0	0.03 ± 0.05	32.19 ± 0.0	0.03 ± 0.04

	bpic2015_5		**sepsis_1**		**sepsis_3**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	4.96 ± 0.0	0.02 ± 0.04	3.08 ± 0.0	0.04 ± 0.04	4.44 ± 0.0	0.04 ± 0.04
cluster_agg	11.24 ± 0.0	0.03 ± 0.05	3.4 ± 0.0	0.04 ± 0.04	5.14 ± 0.0	0.04 ± 0.04
single_agg	7.21 ± 0.0	0.03 ± 0.04	1.75 ± 0.0	0.04 ± 0.04	2.27 ± 0.0	0.04 ± 0.05
prefix_agg	1.1 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.47 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.51 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	1.11 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.47 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.51 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	5.27 ± 0.0	0.03 ± 0.04	2.51 ± 0.0	0.04 ± 0.05	1.52 ± 0.0	0.04 ± 0.04
prefix_index	$\mathbf{0.17 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.12 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.12 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	129.98 ± 0.0	0.03 ± 0.05	8.73 ± 0.0	0.04 ± 0.04	14.11 ± 0.0	0.04 ± 0.04
state_laststate	153.71 ± 0.0	0.03 ± 0.05	9.17 ± 0.0	0.04 ± 0.04	15.53 ± 0.0	0.04 ± 0.05

	bpic2015_1		**production**		**bpic2011_1**	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	9.57 ± 0.0	0.03 ± 0.05	5.35 ± 0.0	0.05 ± 0.05	8.57 ± 0.0	0.05 ± 0.07
cluster_agg	5.93 ± 0.0	0.02 ± 0.04	2.23 ± 0.0	0.05 ± 0.05	9.22 ± 0.0	0.05 ± 0.08
single_agg	2.39 ± 0.0	0.02 ± 0.04	1.48 ± 0.0	0.05 ± 0.05	2.34 ± 0.0	0.04 ± 0.05
prefix_agg	0.74 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.12 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.81 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	0.73 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.12 ± 0.0	$\mathbf{0.0 \pm 0.0}$	0.82 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	3.06 ± 0.0	0.02 ± 0.04	1.45 ± 0.0	0.05 ± 0.04	1.94 ± 0.0	0.04 ± 0.05
prefix_index	$\mathbf{0.12 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.04 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.19 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	74.96 ± 0.0	0.03 ± 0.05	14.28 ± 0.0	0.05 ± 0.04	47.0 ± 0.0	0.04 ± 0.05
state_laststate	84.69 ± 0.0	0.03 ± 0.04	7.58 ± 0.0	0.05 ± 0.04	90.1 ± 0.0	0.05 ± 0.07

	bpic2017_R	
method	training_total (s)	prediction_avg (ms)
cluster_laststate	34.43 ± 0.0	0.03 ± 0.04
cluster_agg	52.84 ± 0.0	0.03 ± 0.04
single_agg	194.26 ± 0.0	0.03 ± 0.04
prefix_agg	6.85 ± 0.0	$\mathbf{0.0 \pm 0.0}$
prefix_laststate	7.66 ± 0.0	$\mathbf{0.0 \pm 0.0}$
single_laststate	57.99 ± 0.0	0.04 ± 0.05
prefix_index	$\mathbf{5.4 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
state_agg	125.27 ± 0.0	0.03 ± 0.04
state_laststate	30.69 ± 0.0	0.03 ± 0.04

Table B.24: Execution times for **inter-case features plus SVM**

	bpic2015_2		bpic2012_C		bpic2015_4	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	2.19 ± 0.0	0.01 ± 0.01	9.6 ± 0.0	0.01 ± 0.01	1.47 ± 0.0	0.01 ± 0.02
cluster_agg	2.16 ± 0.0	0.01 ± 0.02	56.16 ± 0.0	0.01 ± 0.01	1.59 ± 0.0	0.01 ± 0.02
single_agg	4.54 ± 0.0	0.01 ± 0.01	67.56 ± 0.0	**0.0 ± 0.01**	2.55 ± 0.0	0.01 ± 0.01
prefix_agg	0.85 ± 0.0	**0.0 ± 0.0**	2.49 ± 0.0	**0.0 ± 0.0**	0.63 ± 0.0	**0.0 ± 0.0**
prefix_laststate	0.83 ± 0.0	**0.0 ± 0.0**	2.5 ± 0.0	**0.0 ± 0.0**	0.61 ± 0.0	**0.0 ± 0.0**
single_laststate	9.44 ± 0.0	0.01 ± 0.01	57.31 ± 0.0	**0.0 ± 0.01**	1.68 ± 0.0	0.01 ± 0.01
prefix_index	**0.13 ± 0.0**	**0.0 ± 0.0**	**0.7 ± 0.0**	**0.0 ± 0.0**	**0.1 ± 0.0**	**0.0 ± 0.0**
state_agg	4.24 ± 0.0	0.01 ± 0.02	6.45 ± 0.0	0.01 ± 0.01	3.7 ± 0.0	0.01 ± 0.02
state_laststate	3.42 ± 0.0	0.01 ± 0.02	5.07 ± 0.0	**0.0 ± 0.01**	2.97 ± 0.0	0.01 ± 0.02

	bpic2012_D		bpic2017_A		bpic2011_4	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	21.55 ± 0.0	0.01 ± 0.01	2881.8 ± 0.0	0.01 ± 0.02	6.18 ± 0.0	0.02 ± 0.03
cluster_agg	23.2 ± 0.0	0.01 ± 0.01	-	-	4.14 ± 0.0	0.02 ± 0.03
single_agg	254.4 ± 0.0	**0.0 ± 0.01**	-	-	38.75 ± 0.0	0.02 ± 0.03
prefix_agg	2.45 ± 0.0	**0.0 ± 0.0**	6.82 ± 0.0	**0.0 ± 0.0**	1.17 ± 0.0	**0.0 ± 0.0**
prefix_laststate	2.47 ± 0.0	**0.0 ± 0.0**	-	-	1.16 ± 0.0	**0.0 ± 0.0**
single_laststate	41.05 ± 0.0	**0.0 ± 0.01**	-	-	31.59 ± 0.0	0.02 ± 0.03
prefix_index	**0.72 ± 0.0**	**0.0 ± 0.0**	**6.3 ± 0.0**	**0.0 ± 0.0**	**0.19 ± 0.0**	**0.0 ± 0.0**
state_agg	8.04 ± 0.0	0.01 ± 0.01	1007.23 ± 0.0	0.01 ± 0.01	5.53 ± 0.0	0.02 ± 0.03
state_laststate	5.58 ± 0.0	**0.0 ± 0.01**	173.5 ± 0.0	0.01 ± 0.01	4.56 ± 0.0	0.02 ± 0.03

	bpic2012_A		bpic2015_3		bpic2011_3	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	31.49 ± 0.0	0.01 ± 0.01	3.96 ± 0.0	0.01 ± 0.02	2.24 ± 0.0	0.02 ± 0.03
cluster_agg	337.89 ± 0.0	0.01 ± 0.01	5.37 ± 0.0	0.01 ± 0.02	6.23 ± 0.0	0.03 ± 0.04
single_agg	158.94 ± 0.0	**0.0 ± 0.01**	20.77 ± 0.0	0.01 ± 0.01	14.21 ± 0.0	0.02 ± 0.03
prefix_agg	2.5 ± 0.0	**0.0 ± 0.0**	1.36 ± 0.0	**0.0 ± 0.0**	0.6 ± 0.0	**0.0 ± 0.0**
prefix_laststate	2.42 ± 0.0	**0.0 ± 0.0**	1.32 ± 0.0	**0.0 ± 0.0**	0.6 ± 0.0	**0.0 ± 0.0**
single_laststate	433.88 ± 0.0	**0.0 ± 0.01**	117.03 ± 0.0	0.01 ± 0.01	11.69 ± 0.0	0.02 ± 0.03
prefix_index	**0.71 ± 0.0**	**0.0 ± 0.0**	**0.21 ± 0.0**	**0.0 ± 0.0**	**0.18 ± 0.0**	**0.0 ± 0.0**
state_agg	9.19 ± 0.0	0.01 ± 0.01	5.69 ± 0.0	0.01 ± 0.02	3.24 ± 0.0	0.02 ± 0.03
state_laststate	5.24 ± 0.0	**0.0 ± 0.01**	4.74 ± 0.0	0.01 ± 0.02	2.88 ± 0.0	0.02 ± 0.03

	sepsis_2		bpic2011_2		bpic2017_C	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	0.52 ± 0.0	0.02 ± 0.02	4.49 ± 0.0	0.02 ± 0.03	1575.74 ± 0.0	0.01 ± 0.02
cluster_agg	0.41 ± 0.0	0.02 ± 0.02	8.38 ± 0.0	0.02 ± 0.03	-	-
single_agg	1.07 ± 0.0	0.02 ± 0.02	36.44 ± 0.0	0.02 ± 0.03	4659.2 ± 0.0	0.02 ± 0.02
prefix_agg	0.16 ± 0.0	**0.0 ± 0.0**	1.25 ± 0.0	**0.0 ± 0.0**	6.81 ± 0.0	**0.0 ± 0.0**
prefix_laststate	0.16 ± 0.0	**0.0 ± 0.0**	1.22 ± 0.0	**0.0 ± 0.0**	6.79 ± 0.0	**0.0 ± 0.0**
single_laststate	0.3 ± 0.0	0.02 ± 0.02	27.1 ± 0.0	0.02 ± 0.03	4118.18 ± 0.0	0.02 ± 0.02
prefix_index	**0.12 ± 0.0**	**0.0 ± 0.0**	**0.2 ± 0.0**	**0.0 ± 0.0**	**6.19 ± 0.0**	**0.0 ± 0.0**
state_agg	0.78 ± 0.0	0.02 ± 0.02	5.76 ± 0.0	0.02 ± 0.04	-	-
state_laststate	0.73 ± 0.0	0.02 ± 0.02	4.58 ± 0.0	0.02 ± 0.03	-	-

	bpic2015_5		sepsis_1		sepsis_3	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	5.43 ± 0.0	0.01 ± 0.02	0.89 ± 0.0	0.01 ± 0.02	1.48 ± 0.0	0.02 ± 0.02
cluster_agg	4.74 ± 0.0	0.01 ± 0.02	1.4 ± 0.0	0.02 ± 0.02	0.85 ± 0.0	0.02 ± 0.02
single_agg	13.38 ± 0.0	0.01 ± 0.02	2.05 ± 0.0	0.01 ± 0.02	1.15 ± 0.0	0.01 ± 0.02
prefix_agg	1.11 ± 0.0	**0.0 ± 0.0**	0.47 ± 0.0	**0.0 ± 0.0**	0.51 ± 0.0	**0.0 ± 0.0**
prefix_laststate	1.08 ± 0.0	**0.0 ± 0.0**	0.48 ± 0.0	**0.0 ± 0.0**	0.51 ± 0.0	**0.0 ± 0.0**
single_laststate	12.29 ± 0.0	0.01 ± 0.02	0.77 ± 0.0	0.01 ± 0.02	0.87 ± 0.0	0.01 ± 0.02
prefix_index	**0.17 ± 0.0**	**0.0 ± 0.0**	**0.12 ± 0.0**	**0.0 ± 0.0**	**0.12 ± 0.0**	**0.0 ± 0.0**
state_agg	5.84 ± 0.0	0.02 ± 0.02	1.26 ± 0.0	0.02 ± 0.02	1.22 ± 0.0	0.02 ± 0.02
state_laststate	4.58 ± 0.0	0.01 ± 0.02	1.1 ± 0.0	0.01 ± 0.02	1.22 ± 0.0	0.01 ± 0.02

	bpic2015_1		production		bpic2011_1	
method	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)	training_total (s)	prediction_avg (ms)
cluster_laststate	2.29 ± 0.0	0.01 ± 0.02	0.22 ± 0.0	0.01 ± 0.01	5.13 ± 0.0	0.02 ± 0.03
cluster_agg	1.95 ± 0.0	0.01 ± 0.02	0.28 ± 0.0	0.01 ± 0.01	4.44 ± 0.0	0.02 ± 0.04
single_agg	4.97 ± 0.0	0.01 ± 0.01	0.15 ± 0.0	0.01 ± 0.01	21.82 ± 0.0	0.02 ± 0.03
prefix_agg	0.75 ± 0.0	**0.0 ± 0.0**	0.12 ± 0.0	**0.0 ± 0.0**	0.82 ± 0.0	**0.0 ± 0.0**
prefix_laststate	0.72 ± 0.0	**0.0 ± 0.0**	0.12 ± 0.0	**0.0 ± 0.0**	0.81 ± 0.0	**0.0 ± 0.0**
single_laststate	3.67 ± 0.0	0.01 ± 0.01	0.14 ± 0.0	0.01 ± 0.01	18.71 ± 0.0	0.02 ± 0.03
prefix_index	**0.11 ± 0.0**	**0.0 ± 0.0**	**0.04 ± 0.0**	**0.0 ± 0.0**	**0.19 ± 0.0**	**0.0 ± 0.0**
state_agg	4.62 ± 0.0	0.01 ± 0.02	0.43 ± 0.0	0.01 ± 0.01	4.12 ± 0.0	0.02 ± 0.03
state_laststate	3.58 ± 0.0	0.01 ± 0.02	0.36 ± 0.0	0.01 ± 0.01	3.48 ± 0.0	0.02 ± 0.03

	bpic2017_R	
method	training_total (s)	prediction_avg (ms)
cluster_laststate	144.51 ± 0.0	0.01 ± 0.01
cluster_agg	284.54 ± 0.0	0.01 ± 0.01
single_agg	2551.66 ± 0.0	0.01 ± 0.02
prefix_agg	6.96 ± 0.0	**0.0 ± 0.0**
prefix_laststate	6.9 ± 0.0	**0.0 ± 0.0**
single_laststate	-	-
prefix_index	**5.3 ± 0.0**	**0.0 ± 0.0**
state_agg	165.78 ± 0.0	0.01 ± 0.01
state_laststate	-	-