Knowledge and Information Systems (2020) 62:669–686
https://doi.org/10.1007/s10115-019-01365-y

REGULAR PAPER

Anomaly detection of event sequences using multiple temporal resolutions and Markov chains

Martin Boldt[1] · **Anton Borg**[1] · **Selim Ickin**[2] · **Jörgen Gustafsson**[2]

Received: 24 April 2018 / Revised: 4 May 2019 / Accepted: 6 May 2019 / Published online: 15 May 2019

Abstract
Streaming data services, such as video-on-demand, are getting increasingly more popular, and they are expected to account for more than 80% of all Internet traffic in 2020. In this context, it is important for streaming service providers to detect deviations in service requests due to issues or changing end-user behaviors in order to ensure that end-users experience high quality in the provided service. Therefore, in this study we investigate to what extent sequence-based Markov models can be used for anomaly detection by means of the end-users' control sequences in the video streams, i.e., event sequences such as play, pause, resume and stop. This anomaly detection approach is further investigated over three different temporal resolutions in the data, more specifically: 1 h, 1 day and 3 days. The proposed anomaly detection approach supports anomaly detection in ongoing streaming sessions as it recalculates the probability for a specific session to be anomalous for each new streaming control event that is received. Two experiments are used for measuring the potential of the approach, which gives promising results in terms of precision, recall, F_1-score and Jaccard index when compared to k-means clustering of the sessions.

Keywords Anomaly detection · Markov Chains · Multiple temporal resolutions · Event sequences · Video-on-demand

✉ Martin Boldt
martin.boldt@bth.se

Anton Borg
anton.borg@bth.se

Selim Ickin
selim.ickin@ericsson.com

Jörgen Gustafsson
jorgen.gustafsson@ericsson.com

[1] Department of Computer Science and Engineering, Blekinge Institute of Technology, 370 24 Karlskrona, Sweden

[2] Ericsson Research, Machine Intelligence and Automation, 164 40 Stockholm, Sweden

1 Introduction

As streaming data services, such as video-on-demand (VoD), are increasing in popularity, the streaming data service providers are in need of methods to detect issues in the service in order to provide quality service to its customers [12,14–16,28]. Each stream, or session, consists of a sequence of data over time [20], and content of streams vary with the type of streaming data. However, in a VoD context each stream consists of both content and control data, where the latter can be considered a sequence of control events, e.g., start, stop or re-buffering requests. As such, the control sequence is indicative of end-user viewing behaviors, which also reflects problems that the end-users experience in the streaming service, e.g., quality degradation in the data stream. One example is that large numbers of client video re-buffering requests could indicate some issue related to the image quality. As such, the control sequences from VoD clients represent an interesting means of detecting deviations from normal viewing behaviors among the customers, which could be used to highlight for instance quality degradation issues in the service. This problem at hand matches an anomaly detection approach well [4,9].

Anomaly detection in this setting refers to the process of identifying end-users' event sequences that do not conform to the notion of normal behavior. We use Grubbs' definition of an anomaly (or outlier) as an event sequence that is inconsistent to other event sequences in the data sample [10]. By learning from the probability of different control sequences as well as for each individual event in control sequences, it is possible to model normal behavior for VoD sessions [20,26]. Deviant control sequences are thus regarded as anomalous in the stream [7,18].

Various types of Markov models are suitable for classifying sequential data [17,25]. Such models are clearly useful for learning end-user behaviors based on the probabilities of individual events (or states) in event sequences [22], where the probability of any event depends on the previous event in the sequence due to the Markov property. In its simplest form, the Markov property declares that any event is determined by the previous event only, and the first event is determined by each events' prior probability. If one wants to take more than one prior event into account for determining the probability of the next event, then a higher-order Markov model should be used. For instance, to take the previous 3 events into account a Markov model of order 3 should be used.

In the present study, a particular type of Markov models called Markov chains are used, which operate on sequential time-discrete sequences of events. Markov chains can be used to calculate the probability of a future event based on previous events that have been recorded in an ongoing and incomplete sequence. Thus, by using Markov chains in the context of VoD streams it is possible to model normal end-user behavior and then detect anomalies from the normal behavior, which could assist domain experts in identifying problems in the video streaming service [20,26].

This study investigates the use of Markov chains for detecting anomalies in control sequences related to VoD streams. Further, we intend to investigate anomaly detection over multiple temporal resolutions since an anomaly within one timespan (e.g., one hour) might be considered normal behavior when instead viewed within another timespan (e.g., 24 h). So, the goal with this study is to investigate the use of Markov chains over multiple temporal timespans in order to investigate how anomalies in end-users' video streaming behaviors disperse over time.

The remainder of this study is organized as follows. In the next subsections, related work as well as the aim and scope of the study are presented. Then, in the next section the data and its representation are presented, followed by a description of the proposed anomaly detection

approach in Sect. 3. Then, the experiment design is presented in Sect. 4, which is followed by the results in Sect. 5. Finally, the results are discussed in Sect. 6 and concluded in Sect. 7, which also suggests avenues for future work.

1.1 Related work

Padhraic Smyth investigated clustering of event sequences using the transition matrices of hidden Markov models [25]. He further presented a method based on cross-validated likelihoods from the hidden Markov models to determine the number of true clusters in a sequence dataset. Chandola et al. presented an in-depth survey of anomaly detection approaches and the various assumptions that they are based on [4]. The authors presented definitions of anomalies, different approaches to anomaly detection, as well as examples of applications.

The study by Ahmed et al. included an in-depth survey of various clustering-based anomaly detection techniques and a discussion of the limited availability of real-world data and strategies for mitigating this [3]. Emmott et al. presented a meta-analysis of anomaly detection algorithms together with benchmark corpuses that could be used for evaluation and comparison of anomaly detection algorithms [7]. Further, they also evaluated different anomaly detection algorithms using these benchmark corpuses which resulted in that isolation forest was the best performing algorithm in general.

Mori et al. presented a method for behavior patterns extraction and anomaly detection in room sensory data in an elderly smart home context [20]. They used hidden Markov models followed by k-means clustering on the resulting likelihood matrices. The models were able to learn patterns of daily life routines among the elderly persons and could detect anomalies in these routines when mapped to temporal 7x24 matrices representing day-in-the-month by hour-in-the-day. Unfortunately, the authors did not include any internal or external evaluation metrics regarding the accuracy of the models. Related, Hamid et al. discussed anomaly detection in videos using bag-of-events n-grams [11]. The approach allowed detection of anomalies in video surveillance and different sub-classes of behavior.

Event sequences, e.g., network events or clickstreams, have been investigated in a few domains. Chandola et al. provided an extensive survey of anomaly detection for discrete sequences [5]. The authors presented differences between various approaches and the algorithms most often used per approach, as well as their strength and weaknesses. Further, the authors suggested how approaches could be adapted for similar problems. Anomaly detection has been applied by, e.g., Aguiar et al. [1] and Meng et al. [19]. The former was able to predict early exits of users in a video services, based on the clickstream sequences [1]. This provides deeper understanding of the user-base, e.g., when a user is likely to stop watching specific videos. Meng et al. modeled network event sequences as Markov chains in order to build network profiles for different services (e.g., Wifi sessions) [19]. The initial Markov chains were created using clean data, and by then creating new Markov chains for new datasets and comparing them to the clean Markov chains, possible anomalies could be detected as well as their features.

Ahmad and Purdy investigated a method for time-series anomaly detection named hierarchical temporal memory [2]. The model computed the anomaly likelihood based on how different the actual value at a specific position is from the predicted value. The anomaly likelihood is different from a threshold in that it checks if the value belongs to the same distribution as the values within a certain time window. As such, the approach is capable of handling changes in the data over time. This, however, requires a time series longer than the time window.

Balasingam et al. presented a VoD anomaly detection approach. The authors suggested using principal component analysis (PCA) to detect groups of features that correspond to normal behavior and anomalous behavior. The data were clustered into a number of groups based on the PCA, and multivariate Gaussian modeling for each cluster. The steps were then repeated as required by changes in the data. Based on this, an online anomaly detection framework was constructed. However, while the approach allowed multiple anomalies to be detected, the interpreting PCA output was often manual. Further, the approach for detecting anomalies could possibly be more efficiently implemented using hidden Markov models.

Based on the existing research summarized above, we have identified a research gap related to the analysis of event sequences using Markov chains over multiple temporal resolutions.

1.2 Motivation for the study

The motivation for investigating the effect of anomaly detection over multiple temporal resolutions is twofold: first, in order to decrease the dependability of the anomaly detection capability on the data internals. That is to make the anomaly detection more resistant against various seasonality aspects in the data by detecting anomalies at different temporal resolutions. The second is to use the different temporal resolutions as a means for decreasing the number of false positives returned to the domain expert(s) using the anomaly detection solution.

1.3 Aims & scope of study

The aim of the present study is to investigate to what extent Markov chains can be used to learn the probability of control sequences over different temporal resolutions, and then classify control sequences in order to detect anomalous behavior. VoD streaming data for a random month from a European streaming service provider are used in this study. In the study, batch detection of anomalies is investigated rather than early detection in sessions. However, as stated earlier the latter is supported by the technique.

1.4 Assumptions

This study uses a cluster-based outlier detection method for the initial labeling of anomalies [13]. However, this clustering is not a contribution of the paper. Multiple approaches for using clustering to detect anomalies exists, e.g., outliers within clusters. Consequently, the initial labeling might be preexisting or be done using the most suitable approach.

2 Data & data representation

In this section, the data and how it has been represented are presented. Further, the temporal resolutions used when representing the data are described in more detailed, as are possible privacy concerns of end-users. The dataset constructed, using the data representation described in Sect. 2.1, is later used to fit Markov chain models to the data, as described in Sect. 3.3.

The data consist of complete VoD control streams, where each control stream is regarded as an *instance*. Each instance also consists of a set of sparse *features*: session id, event type and

time stamps are present in all instances. Further, instances can also contain OS information, client id, as well as multiple other features. Of these data, a subsample has been extracted consisting of instances related to a random VoD streaming channel collected during a random month's time in 2017. In total, 104,135 instances were included in this study.

2.1 Data representation

The data were grouped by the session such that an instance contains the complete control sequences related to a specific VoD streaming session. The features available in each instance are session id, event labels, time stamps for the events, and OS information. There are a total of 28 unique event labels (e.g., play, bit rate change, re-buffering, etc.) that control sequences consist of. Instances usually start with some initialization events managing access control of the content and setup of a video stream, which then are followed by various events that adjust the bit rate or video resolution, pauses or fast forwards the video stream, etc. Instances stop with either a playback completed event, or an abort or error event.

Event labels were originally in string format, but were transformed into numeric values (each value corresponding to a unique event label). After the transformation, each instance had the following format:

$$SessionId,\ OS,\ [EventId_0 \ldots EventId_n], [Timestamp_0 \ldots Timestamp_n]$$

The list of EventIds as such can be considered an event sequence. The event sequence is highly related to the concept of clickstreams, that is, a record of how end-users have interacted with a service [1]. In this case, the interaction is not limited to the explicit actions by the end-user, but also includes interactions initiated by clients' applications. However, interactions initiated by clients' applications can be regarded as implicit end-user behavior. As an example, say an end-user selects a too high video resolution given the available bandwidth. Then, the application as a result will request a bit rate change. Thus, the application's interaction to change the bit rate can be seen as an implicit action caused by the end-user's choice of a too high resolution.

2.2 Temporal resolutions

The choice of temporal resolution is a balance between including fine-grained resolutions (e.g., one hour) in order to detect anomalies more quickly on one hand, and detecting more stable anomalies in the data by including lower resolutions (e.g., 3 days). Also, more fine-grained temporal resolutions typically also result in limited amount of data for the models, which negatively affects the capability of learning users' behaviors.

Three temporal resolutions, or time windows, are used in this study. They are described as $t - h$, where t is the point in time that is considered current, and h is the number of prior hours that constitutes the time window that is interesting in the specific resolution. h is, in this study defined as $h \in \{1, 24, 72\}$, i.e., the time windows of interest are 1 h, 1 day, and 3 days. The specific temporal resolutions can be adjusted for domain-specific requirements, and in this study, the three temporal resolutions were chosen after discussions with domain experts working with VoD streaming.

In this study, 30 datasets (10 for each of the three temporal resolutions) were created as a function of t, where the different temporal resolutions per t result in subsets of each other, i.e., $d_{t,1} \in d_{t,24} \in d_{t,72}$. This was done by first randomly sampling 10 3-day datasets that each consists of three sequential days, i.e., 10 datasets where the sessions started within 72 hours

Table 1 Descriptive statistics for the datasets consisting of minimum, median, inter-quartile range (IQR) that is the difference between the 3rd and 1st quartiles, and finally the maximum size

	No. datasets	Min	Median	IQR	Max
Hour datasets	10	332	399	106	797
1-day datasets	10	13,819	14,989	1834	16,806
3-day datasets	10	16,807	60,694	3537	62,818

from t. Next, from each $d_{t,72}$, the $d_{t,24}$ and $d_{t,1}$ were created. This resulted in 10 datasets with a temporal resolution $h = 24$ and 10 datasets with a temporal resolution $h = 1$. So, to summarize there are 30 unique datasets, but all of them could be divided into groups of three where any one-hour dataset is a subset of one of the one-day datasets, which in turn is a subset of one of the three-day datasets.

As shown in Table 1, all three temporal resolutions contain 10 datasets each. The median size for the 3-day datasets is 60,694 instances with an inter-quartile range (IQR) spread of 3537. For the one-day and hour datasets, the median sizes are 14,989 and 399, with IQR spreads of 1834 and 106, respectively.

2.3 End-user privacy concerns

Although the SessionIds within instances could be used to map a session to end-users' accounts, there were no such data available during the present study. Nor were any attempts made in order to map VoD sessions to individual end-users. However, preserving the possibility to do such session to end-user mappings is important for the VoD streaming service provider. Without such possibility, it would not be possible to mitigate identified anomalies by applying targeted actions within the streaming service.

3 Proposed anomaly detection approach

This section presents the proposed anomaly detection approach, the assumptions that it rests on, and how the resulting anomalies can be ranked based on their relative importance.

3.1 Setting the scene

For each instance, a 28×28 transition matrix is created that consists of the transition probabilities of going from each event that exists in a sequence to any other in the given event alphabet. Transitions between two events that never occur in an instance have a probability of 0, while all other transitions have probabilities greater than 0. The transition matrix is a stochastic matrix, which means that the entries in rows related to events that exist in an instance add up to exactly 1.0. This is because they represent a probability distribution for going from one specific event to any other event.

3.2 Initial labeling

Since there exist no labels in the dataset, an initial labeling is necessary to identify anomalous instances. The proposed approach relies on clustering the instances and then using cluster

labels as end states in the Markov chains as explained later in this subsection. The first step is to cluster the instances per dataset using their transition matrices with a cluster-based outlier detection method. As mentioned previously the k-means clustering algorithm is used in this study [8]. The reason for choosing the k-means algorithm is twofold. First, it has been used in previous studies in similar problem domains, e.g., Yasami et al. [29], Muniyandi et al. [21] and Mori et al. [20]. Secondly, k-means is well suited for the problem at hand [5] and has commonly available implementations [23,27]. It should, however, be noted that the anomaly detection approach presented in the present study is technologically independent from the k-means clustering algorithm. Thus, k-means could be replaced with another clustering algorithm if that would be preferable, e.g., given other domain or data specific requirements.

For each dataset, the k-means algorithm divides the instances into distinct clusters based on their internal event sequence similarities represented by the probabilities in the transition matrices. The number of clusters, i.e., k, was in the interval $[2, 20]$ (inclusive). The lower bound for this interval was chosen because the proposed approach requires at least a rudimentary division of the data, i.e., two clusters, while the upper limit should reflect the available hardware configuration available as it limits for how long the clustering process is executing in worst case. The outcome from this step is that all instances in each dataset belong to exactly one of the clusters associated with each particular dataset.

After the clusters are established, a cluster label is added as an end state (also known as an absorbing state) in each instance's event sequence. This means that all instances in the same cluster have the same end state in the event sequence. As an example, all instances in the first cluster will have end state "$Cluster_1$," while all instances in the n:th cluster have end state "$Cluster_n$." Next, the prior probabilities for instances to belong to a particular cluster are calculated as the fraction of instances that are included in each individual cluster divided by the total number of instances in that particular dataset. Finally, the instances in the smallest cluster, i.e., the cluster with fewest instances, are suggested to be regarded as anomalies within the current dataset, similarly to the approach used by He et al. [13].

3.3 Fitting Markov chains to the data

A Markov chain essentially is a transition matrix that holds the transition probabilities for going from one event to any other event in the alphabet. As such, a key aspect when creating a Markov chain is to learn the transition probabilities from some historical data, or to fit the Markov chain to the data. In this study, we fit one Markov chain per dataset, i.e., we learn the transition probabilities for changing from one state to any other state in the alphabet from all instances in a specific dataset. The result from this is that there is exactly one Markov chain, each with its own transition probability matrix, associated with each dataset.

3.4 Anomaly detection of event sequences

There exists a subset of Markov chains called *absorbing* Markov chains [17] that apart from ordinary Markov chains have absorbing states. The absorbing states can be reached from any other state, but once reached, they can never be left. In the proposed anomaly detection approach, the cluster labels that were appended to all event sequences are used as absorbing states. The absorbing states enable the Markov chains to calculate the probability that a particular sequence will end in a particular of the clusters of that dataset, e.g., the anomaly cluster. so, based on the intricate patterns of events in an instance's event sequence a Markov

chain can calculate the probabilities for that instance to belong to each of the clusters related to that dataset.

Once a Markov chain has been fitted to instances in a dataset, it can also be used to carry out anomaly detection for each active/ongoing event sequences that are not complete. In the VoD setting, this means that the probability for an ongoing VoD session being an anomaly can be recalculated for each new VoD event that is received. This also allows domain experts to specify at which probability thresholds they want to be notified about the anomalous session.

3.5 Analysis of anomalies in various temporal resolutions

The last part of the proposed anomaly detection approach involves the investigation of which anomalies that exist in each of the different temporal resolutions. If all instances identified as anomalies in any of the temporal resolutions are considered, there will be quite a few number of instances. By instead considering instances that are identified in two different temporal resolutions, it is possible to reduce the number of identified anomalies. Further, by only considering those instance identified as anomalies in all temporal resolutions it is possible to reduce the number even further, as shown in Results section. Thus, by choosing which temporal resolutions that are considered it is possible to significantly decrease the number of anomalies that are interesting for domain experts to analyze further.

Finally, it is also possible to rank the instances highlighted as anomalies. One possibility could be to rank instances identified as anomalies in all temporal resolutions with the highest rank, while instances identified as anomalies in all but one temporal resolution are ranked with a lower rank, and instances identified as an anomaly in only one temporal resolution are ranked with the lowest rank. However, the exact ranking strategy is dependent of the application, and thus, it needs to be adjusted to the problem that is investigated. For example, there might be cases where anomalies detected in the 1-h resolution are indicative of novel behavior, and as such should be considered.

4 Method

This section describes the method used with regard to metrics, experiment design and statistical tests used for the evaluation of the investigated anomaly detection approach.

4.1 Evaluation metrics

In this study, the positive condition (P) represents anomalies and the negative condition (N) is normal behavior. Instances that are correctly classified as anomalies are true positives (TP) or hits, while correctly classified normal instances are true negatives (TN) or misses. Similarly, incorrectly classified normal instances as anomalies are false positives (FP) or false hits, while incorrectly classified anomalies as normal instances are false negatives (FN) or false misses [8]. These four measures allow the calculation of precision and recall, F_1-score and accuracy [8,27], where F_1-score is the primary metric used in this study. Precision is defined as $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$. It is a measurement of how dispersed the positives are. Recall is defined as $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$, and it shows how many of the positives are actually classified as TP. F_1-score is the harmonic mean between the precision and recall and is calculated as $\frac{2*\mathrm{TP}}{2*\mathrm{TP+FP+FN}}$. Accuracy is defined as $\frac{\mathrm{TP+TN}}{\mathrm{TP+FP+TN+FN}}$, i.e., the fraction of correctly classified instances over all instances.

In addition to the four metrics described above, the Jaccard index is also used for comparing the number of instances classified as anomalies over the three different temporal resolutions. The Jaccard index is calculated as $\frac{A \cap B}{A \cup B}$, where a high Jaccard index indicates that a large portion of instances overlap between the temporal resolutions, while a lower Jaccard index indicates that fewer instances overlap.

4.2 Experiment setup

This study consists of two experiments that are described individually below. The first experiment investigates the Markov chain models ability to pick up anomalous event sequences, while the second experiment investigates the overlap of anomalies between the three temporal resolutions.

4.2.1 Experiment 1: anomaly detection per temporal resolution

This experiment aims to investigate the classification performance of the proposed approach trained on the collected data. In the first experiment, the instances in each dataset were divided into train and test data using random stratified sampling with a 70/30 split [24,27], i.e., 70% of the instances were used for training the model and the remaining 30% were used for testing it. This was repeated 10 times for each temporal resolution. As a result, there were 30 training and 30 corresponding test datasets, i.e., 10 for each temporal resolution described in Sect. 2.2. Then, each training dataset was used for fitting 30 Markov chains as described in Sect. 3. Next, the Markov chains were used for classifying the instances within the corresponding test datasets. This included identifying instances in the minority cluster as anomalies for each of the three temporal resolutions investigated. Finally, the evaluation metrics described above were used to measure the anomaly classification performance for each of the 30 models. To summarize, the independent variable of the experiment has three levels (the three temporal resolutions) and the four evaluation metrics constitute the dependent variables, with F_1-score as primary.

4.2.2 Experiment 2: anomalies in temporal resolution overlaps

In the second experiment, the overlap of anomalies between the three temporal resolutions were measured. This is interesting to investigate as the anomalies could be ranked based on how many temporal resolutions the are identified in. The experiment was conducted on the classifications from experiment 1 using the following four comparisons: 1 day versus 1 h, 3 days versus 1 day, 3 days versus 1 h and finally a comparison between all three temporal resolutions. For each comparison, the Jaccard index was used as the evaluation metric. Thus, the independent variable in this experiment was the three temporal resolutions and the dependent variable was the Jaccard index.

4.3 Statistical approach

All results are described using conventional descriptive statistics such as mean with standard deviation within parentheses. Results are either presented in tables or visualized using box plots that show the spread of the data by quartiles. Cohen's d is used as the measure of effect size where differences between results need to be quantified [6]. It calculates the difference

Table 2 Mean evaluation metrics per temporal resolution with standard deviations within parentheses

	3 days	1 day	1 h	Overall
Precision	0.5737 (0.21)	0.6267 (0.10)	0.5760 (0.08)	0.5921 (0.14)
Recall	0.9507 (0.16)	1.0000 (0.00)	0.9498 (0.16)	0.9668 (0.13)
F_1-score	0.7018 (0.20)	0.7659 (0.08)	0.7114 (0.09)	0.7264 (0.14)
Accuracy	0.6046 (0.16)	0.6267 (0.10)	0.5759 (0.08)	0.6024 (0.12)

between data points from two groups by dividing the difference in group means by the pooled standard deviation of both groups. Cohen's rules of thumb suggest that $d = 0.2$ is a small effect size, while $d = 0.5$ and $d = 0.8$ are medium and large effect sizes, respectively. So, if two groups differ by $d < 0.2$, the difference is trivial, even if it is statistically significant.

4.4 Tools and environment

Experiments, statistical analysis and visualization were managed in `R`,[1] a free software environment for statistical computing and graphics. Important packages used were `Clickstream`, `ClusterR` and `Ggplot2`. All experiments were executed in Linux Debian, while data pre-processing was handled in a SPARC environment using Python.

5 Results

This section presents the results for the two experiments described in the previous section.

5.1 Experiment 1: anomaly detection per temporal resolution

Experiment 1 investigated the classification performance of the proposed approach. Table 2 shows the mean and standard deviation for the four evaluation metrics for the instances in the three temporal resolutions, as well as for all instances in each dataset. Precision ranges between 0.574 and 0.627 with the best performance for the one-day temporal resolution. The difference between the worst and best performing temporal resolution was according to Cohen's d was 0.322, which translates to a small difference. The recall metric ranges between 0.950 and 1.000 with the best performance also for the one-day temporal resolution. The difference according to Cohen's d was 0.443 which also is a small difference. When both precision and recall are averaged using the harmonic mean, the resulting F_1-score ranges 0.702–0.766 with a Cohen's d of 0.420. Finally, the accuracy ranges 0.605–0.627 with a small difference of 0.164 according to Cohen's d.

As can be seen in Table 2, the average F1-score over all temporal resolutions is 0.726 (0.14). Although not an excellent F_1-score, this indicates that the proposed approach is indeed able to pick up interesting patterns in the data over all investigated temporal resolutions.

All four metrics are consistently highest for the one-day temporal resolution, which indicates that the Markov chains trained at the one-day temporal resolution are best suited for learning the event sequences in the unseen test instances.

[1] R version 3.4.3. URL: https://www.r-project.org.

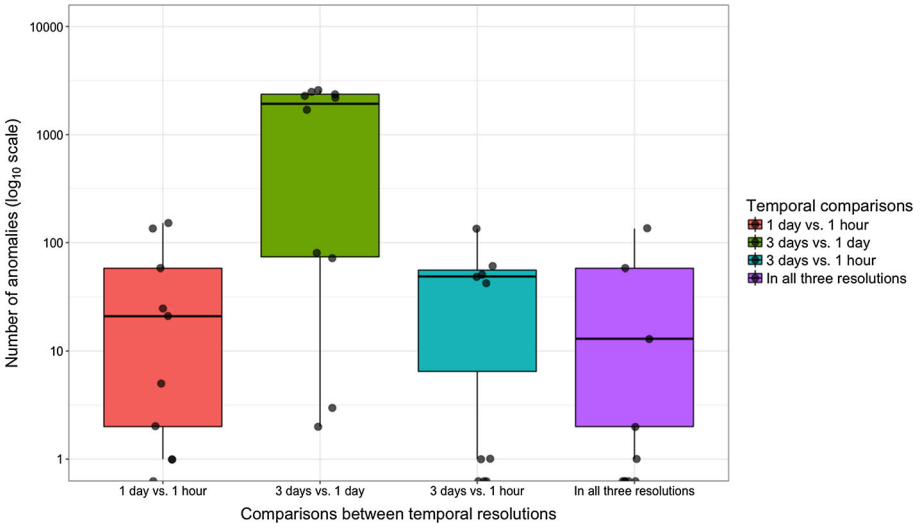

Fig. 1 Box plot showing the number of anomalies that overlap the various temporal resolutions

5.2 Experiment 2: anomalies in temporal resolution overlaps

Experiment 2 investigates the overlap of the classifications from Experiment 1 between the different temporal resolutions.

The box plot in Fig. 1 visualizes the number of overlapping anomalies between the three temporal resolutions, which each contain 10 datasets. As shown, these 10 datasets are represented as dots in the box plot, and the mean number of overlapping anomalies between the one-day and one-hour resolutions is 13. That is, for half out of the 10 datasets there are less than 21 overlapping anomalies, which is a fair number possible for domain experts to analyze manually. Similarly, the median number of anomalies that overlap the three-day and one-hour datasets is 22. However, between the three-day and one-day datasets, the median number of anomalies is instead 1942, which is harder to analyze manually. The number of anomalies that overlap all three temporal resolutions is also shown in Fig. 1 with a median of 11 anomalies, i.e., also a fair number that could be highlighted to domain experts for further analysis given by the low Jaccard index values.

Finally, the Jaccard index is calculated for quantifying the overlap of anomalies between different temporal resolutions. As shown in Fig. 2, only a small fraction of the suggested anomalies overlaps two or more temporal resolutions, which is also shown by the overall low Jaccard index of 5.236×10^{-5}. The exception is the comparison between the three-day and one-day temporal resolutions, which result in four Jaccard measures around 0.2 and a single value at 0.998. To summarize, the low Jaccard indices suggest that only a fraction of all suggested anomalies are consistently predicted to be anomalies over different temporal resolutions.

6 Discussion

Anomalous event sequences could be the result of new or updated software, disruptions in the networks, highly popular video streams, etc. So, identifying such anomalies is important

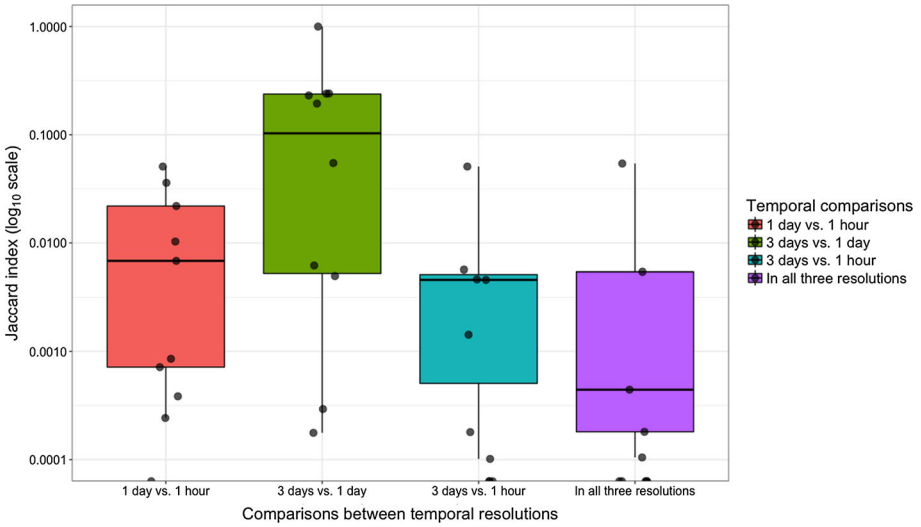

Fig. 2 Box plot showing the Jaccard index between the various temporal resolutions

Table 3 Anomaly cases based on prediction for different temporal resolutions

	1-h	1-day	3-days	Weight
Normal behavior				–
Anomaly, case 1			*	Weak
Anomaly, case 2		*		Weak
Anomaly, case 3	*			Weak
Anomaly, case 4		*	*	Medium
Anomaly, case 5	*		*	Medium
Anomaly, case 6	*	*		Medium
Anomaly, case 7	*	*	*	Strong

*Instances classified as an anomaly within the specific temporal resolution

for the service providers in order to maintain a high quality of service. Identifying anomalous event sequences allows service providers to respond by various means, e.g., using roll-back software for faulty updates. The impact from such problems might not be large enough to make a noticeable difference on the service quality for individual end-users, but might have large aggregated negative effects on the service provider. Therefore, more user-oriented detection methods, e.g., end-user surveys and feedback comments, might not be ideal although such methods could be used in parallel to the proposed approach.

The proposed approach investigates to what extent Markov chains trained for specific temporal resolutions classify anomalies differently, e.g., what is considered an anomaly for the one-hour temporal resolution might be different for the three-day temporal resolution. Detecting anomalies over different temporal resolutions results in different cases depending on which resolution an anomaly is identified in. Three temporal resolutions result in the eight (2^3) different cases enumerated in Table 3.

Case 1–3 represent anomalies detected in only one temporal resolution. These anomalies are susceptible to seasonality, in the sense that they are dependent on the data used to train the

Table 4 Five VoD streaming sessions (labeled S1–S5) showing their first five streaming events and whether the Markov chains (from left to right the 1-h, 1-day, and 3-day models, respectively) regard them as normal (*N*) or anomalies (**A**) after each new event that arrives sequentially in each session

	Event 1	Event 2	Event 3	Event 4	Event 5	...
S1	Granted	PlayCreated	Handshake	InitComplete	PlayStarted	
	{N,N,N}	*{N,N,N}*	*{N,N,N}*	*{N,N,N}*	*{N,N,N}*	...
S2	Granted	PlayReady	PlayCreated	BuffStart	Handshake	
	{N,N,N}	*{N,N,N}*	*{N,N,N}*	*{N,N,N}*	{*N*,*N*,***A***}	...
S3	Granted	Resume	PlayCreated	BitrateChange	InitComplete	
	{N,N,N}	{***A***,*N*,***A***}	{***A***,***A***,***A***}	{***A***,***A***,***A***}	{***A***,***A***,***A***}	...
S4	Heartbeat	Heartbeat	Paused	Aborted	Completed	
	{*N*,***A***,***A***}	{*N*,***A***,***A***}	{*N*,***A***,***A***}	{*N*,***A***,***A***}	{***A***,***A***,***A***}	...
S5	BuffStart	BuffStop	ScrubTo	Heartbeat	ScrubTo	
	{***A***,***A***,*N*}	{***A***,***A***,*N*}	{***A***,***A***,***A***}	{***A***,***A***,***A***}	{***A***,***A***,***A***}	...

Markov chain for the specific temporal resolution. The existing anomaly detection research has analyzed anomalies in single temporal resolutions, which this study aims to extend. Case 4–6 concern anomalies that have been classified as such in more than one temporal resolution, i.e., it is less likely that seasonality has affected the classification. Case 7 concerns the strongest classification, in the sense that instances must be identified as an anomaly in all three temporal resolutions. This means that the event patterns in those anomalous instances in the 1-h, 1-day and 3-days datasets all are distinctly different to the other instances in these datasets, respectively. A more concrete example is discussed in relation to Table 4 below. It should be noted that case 7 therefore indicates a stronger anomaly, as all Markov chain models trained on different temporal resolutions agree on the classification.

In the same sense, the anomalies in case 1–3 can be considered weaker. However, the anomalies in case 1–3 might still be important depending on the underlying problems that are investigated (e.g., novel anomalies). Further, when a higher agreement is required between the Markov chains, the risk for true negatives also increase. As such, depending on the problem this proposed anomaly detection approach could be used to alleviate the level of agreement should be considered carefully. Furthermore, the higher agreement required the fewer anomalies will be detected (as can be seen in Fig. 1).

As mentioned previously, the Markov chains are capable of predicting the probability for a session to be an anomaly for each event that arrives in ongoing sessions. Table 4 shows a sample of five sessions and whether three Markov chain models from each of the three different temporal resolutions regard the sessions as normal or anomalies based on the first five events in each session. A sample of anomalies, such as those in Table 4, were shown to domain experts, and they regarded the detected event sequences at the different temporal resolutions to be very interesting. As these types of identified patterns could help detect potential problems related to the use of the service.

For instance, in the top row the first five events of a normal session (labeled S1) are as follows: (1) granted access to video content, (2) playback session created, (3) handshake performed, (4) initialization complete, and (5) video playback started. Sessions S2–S5 in Table 4 are regarded as anomalies at various degrees by the Markov chains. Session S2 is probably regarded as an anomaly because of a misplaced handshake event, but this is only

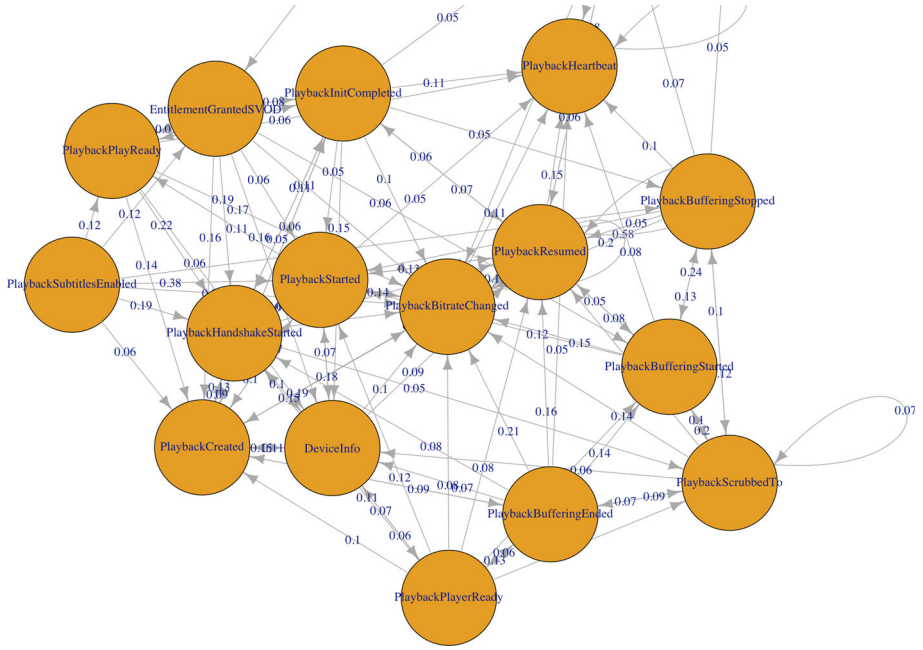

Fig. 3 A subset of a graph representation of one of the Markov chain models, where nodes represent session events and directed edges between them represent transition probabilities

detected by the Markov chains trained on 3 days' worth of data while the 1 h and 1 day do not regard the session as an anomaly.

Session S3 is regarded an anomaly by both the 1-h and 3-day models due to that the second event is a resume playback event, which is not proceeded by session creation and initialization events. However, the 1-day Markov chain does not regard that as an anomaly. Thus, it can be concluded that resume playback events that are proceeded by grant access events are less common in both the 1-h and 3-day temporal resolutions, while such event-pairs are more common during the 1-day resolution and therefore not detected as anomalies by the 1-day model.

Finally, sessions S4 and S5 are both detected as anomalies by all three models after 5 and 3 events, respectively, and before that a majority of the three models regard the sessions as anomalies as well. Both sessions are probably regarded as anomalies in the first event because they lack an initial access-granted event. However, the fact that the 1-h Markov chain does not regard S4 as an anomaly indicates that there are an increased number such sessions without access-granted event, when compared to the 1-day and 3-day temporal resolutions. Session S5 shows a somewhat similar pattern but with the 3-day model differing from the other two models.

Identifying these kinds of session anomalies in the data and highlighting them to domain experts are useful for various reasons, for instance, as a way to detect non-obvious event sequences, and their occurrence in different temporal resolutions, that aid domain experts in detecting strange behavior in the streaming service. Thus, detecting such anomalous sessions could give the domain experts better insight, which in turn can lead to better service quality due to the detection of strange behaviors such as misconfigurations and misuse.

Finally, Fig. 3 depicts a graph representation of one of the Markov chain models. Even though the graph is a subset of the whole graph (only 16 out of the total 28 events) that contains probability edges with values higher than 0.05, it is still quite complex with many interchanging edges. However, the graph still shows how the internal working of Markov chain models can be represented in an appealing fashion that allow domain experts to understand their inner decision logic. Other representations of models' inner decision logics are also possible, one example being heat maps based on the Markov chains' transition tables.

7 Conclusion & future work

This paper presents an anomaly detection method for time-discrete event sequences based on Markov chain models that analyze the existing sequence data using different temporal resolutions. Two experiments show interesting results for the Markov chains when compared to an initial k-means clustering as the F_1-score ranges 0.701 and 0.766 for the investigated temporal resolutions. Further, by focusing on anomalies identified in two, or all three, temporal resolutions it is possible to reduce the number of anomalies that are highlighted to the domain experts for further analysis. The Markov chains for the three temporal resolutions also reveal how anomalies disperse in time given these temporal resolutions.

As future work, it would be interesting to investigate longer temporal intervals for the sake of capturing seasonality in the data, e.g., a week, month and year. Further, it would be interesting to study additional strategies for determining which clusters are anomalies, e.g., by measuring the mean distance between instances and the centroid. Finally, it would be interesting to evaluate the method on synthetic data as well as collect labeled data by means of domain experts in order to detect which sessions they regard as most interesting in their field of work.

Acknowledgements This research work is supported by the Knowledge Foundation in Sweden through project "Scalable resource-efficient systems for big data analytics" (Grant: 20140032).

References

1. Aguiar E, Nagrecha S, Chawla NV (2015) Predicting online video engagement using clickstreams. In: 2015 IEEE international conference on data science and advanced analytics (DSAA), pp 1–10. https://doi.org/10.1109/DSAA.2015.7344873
2. Ahmad S, Purdy S (2016) Real-time anomaly detection for streaming analytics. arXiv:1607.02480v1
3. Ahmed M, Mahmood AN, Islam MR (2016) A survey of anomaly detection techniques in financial domain. Future Gener Comput Syst 55:278–288
4. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv (CSUR) 41(3):15:1–15:58. https://doi.org/10.1145/1541880.1541882
5. Chandola V, Banerjee A, Kumar V (2012) Anomaly detection for discrete sequences: a survey. IEEE Trans Knowl Data Eng 24(5):823–839
6. Cohen J (1988) Statistical power analysis for the behavioral sciences, 2nd edn. Lawrence Earlbaum associates, Hillsdale
7. Emmott A, Das S, Dietterich T, Fern A, Wong WK (2015) A meta-analysis of the anomaly detection problem. arXiv:1503.01158v2

8. Flach P (2012) Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press, Cambridge
9. Fox AJ (1972) Outliers in time series. J R Stat Soc 34(3):350–363
10. Grubbs FE (1969) Procedures for detecting outlying observations in samples. Technometrics 11(1):1–21
11. Hamid R, Johnson A, Batta S, Bobick A, Isbell C, Coleman G (2005) Detection and explanation of anomalous activities: representing activities as bags of event n-grams. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). IEEE, pp 1031–1038
12. Hareesh K, Manjaiah HD (2011) Peer-to-peer live streaming and video on demand design issues and its challenges. arXiv:1111.6735v1
13. He Z, Xu X, Deng S (2003) Discovering cluster-based local outliers. Pattern Recognit Lett 24(9):1641–1650. https://doi.org/10.1016/S0167-8655(03)00003-5
14. Hosseini M (2017) A survey on cloud video multicasting over mobile networks. arXiv:1707.00239v1
15. Kanrar S (2012) Analysis and implementation of the large scale video-on-demand system. Int J Appl Inf Syst 1(4):41–49. arXiv:1202.5094v1
16. Kanrar S, Mandal NK (2017) Approximation of bandwidth for the interactive operation in video on demand system. In: International conference on IoT in social, mobile, analytics and cloud (I-SMAC), pp. 343–346. https://doi.org/10.1109/I-SMAC.2017.8058368
17. Kemeny JG, Snell JL (1960) Finite Markov chains. Springer, Princeton
18. Ma J, Le S, Wang H, Zhang Y, Aickelin U (2016) Supervised anomaly detection in uncertain pseudoperiodic data streams. ACM Trans Internet Technol 16(1):4:1–4:20. https://doi.org/10.1145/2806890
19. Meng X, Jiang G, Zhang H, Chen H, Yoshihira K (2008) Automatic profiling of network event sequences: algorithm and applications. In: IEEE INFOCOM 2008—IEEE conference on computer communications. IEEE, pp 266–270
20. Mori T, Fujii A, Shimosaka M, Noguchi H, Sato T (2007) Typical behavior patterns extraction and anomaly detection algorithm based on accumulated home sensor data. In: Future generation communication and networking (FGCN). IEEE, pp 12–18
21. Muniyandi AP, Rajeswari R, Rajaram R (2012) Network anomaly detection by cascading k-means clustering and c4.5 decision tree algorithm. Proc Eng 30:174–182. https://doi.org/10.1016/j.proeng.2012.01.849
22. Rabiner L, Juang B (1986) An introduction to hidden Markov models. IEEE ASSP Mag 3(1):4–16. https://doi.org/10.1109/MASSP.1986.1165342
23. Scholz M (2016) R package clickstream: analyzing clickstream data with Markov chains. J Stat Softw 74(4):1–17. https://doi.org/10.18637/jss.v074.i04
24. Sheskin D (2007) Handbook of parametric and nonparametric statistical procedures. Chapman & Hall, Boca Raton
25. Smyth P (1997) Clustering sequences with hidden Markov models. Advances in neural information processing systems. MIT Press, Cambridge, pp 648–654
26. Wang F, Zhu H, Tian B, Xin Y, Niu X, Yang Y (2011) A HMM-based method for anomaly detection. In: Multimedia technology (IC-BNMT 2011). IEEE, pp 276–280
27. Witten I, Frank E, Hall M (2011) Data mining—practical machine learning tools and techniques, 3rd edn. Elsevier, Burlington
28. Wu J, Zhou Y, Chiu DM, Zhu Z (2016) Modeling dynamics of online video popularity. IEEE Trans Multimed 18(9):1882–1895. https://doi.org/10.1109/TMM.2016.2579600
29. Yasami Y, Mozaffari SP (2010) A novel unsupervised classification approach for network anomaly detection by k-means clustering and id3 decision tree learning methods. J Supercomput 53(1):231–245. https://doi.org/10.1007/s11227-009-0338-x

Martin Boldt (Ph.D.) is a senior lecturer at the Department of Computer Science at Blekinge Institute of Technology in Sweden. Martin's research interest is in data science methods, which often are applied in network and information security contexts. Recent studies revolve around anomaly detection, failure prediction and sentiment analysis in various network settings. In prior research projects, he has cooperated with law enforcement agencies in applications related to computational criminology. He was general chair for IEEE EISIC'18 and a member of the program committee for conferences. Martin reviews manuscripts for various journals and conferences, such as KAIS, ICML, NIPS and UAI.

Anton Borg (Ph.D.) is a senior lecturer at Blekinge Institute of Technology in Sweden. His research interests are applied machine learning, and increasingly data science, methods. He has worked on anomaly detection, document classification, and sentiment analysis in telecommunication settings. Other research interests have included how machine learning can be applied in a law enforcement setting, as well as computer security. Anton was general chair for EISIC'18. Further, Anton has reviewed manuscripts for various journals and conferences.

Selim Ickin (Ph.D.) is a senior researcher in the area of machine learning at Ericsson Research in Sweden. His recent research interest is distributed machine learning and works highly toward intelligent software prototyping. He has worked in numerous data-driven machine learning projects in diverse domains targeting to improve network-based mobile application performance; to reduce subscriber churn rate for a video service provider; and to reduce network operations cost by applying various machine learning techniques. He has contributed to highly ranked international conferences and journal articles since 2010. He has also a few patents in the area of machine learning on network-based applications.

Jörgen Gustafsson is a research leader at Ericsson Research in Stockholm, Sweden. His research interest includes artificial intelligence-oriented solutions in Telecom systems. He has managed several Telecom research projects in various problem domains, e.g., relating to anomaly detection, fault prediction, distributed learning, and deep and reinforcement learning. He has a degree in computer science from Linköping University.