Expert Systems

WILEY

Expert Systems

ORIGINAL ARTICLE OPEN ACCESS

Cognitive Based Detection of Anomalous Sequences Using Bayesian Surprise

Ken McGarry | David Nelson

School of Computer Science and Engineering, Faculty of Business and Technology, University of Sunderland, Sunderland, UK

Correspondence: Ken McGarry (ken.mcgarry@sunderland.ac.uk)

Received: 3 January 2023 | **Revised:** 14 July 2025 | **Accepted:** 15 July 2025

Keywords: anomaly | Bayesian surprise | entropy | interestingness measure | probabilistic suffix tree | sequences

ABSTRACT

In this work we implement Bayesian surprise as a method to sift through sequences of discrete patterns and identify any unusual or interesting patterns that deviate from known sequences. Surprise is a biological trait inherent in humans and animals and is essential for many creative acts and efforts of discovery. Numerous technical domains are comprised of discrete elements in sequences such as e-commerce transactions, genome data searching, online financial transactions of many types, criminal cyber-attacks and life-course data from sociology. In addition to the complexity and computational burden of this type of problem is the issue of their rarity. Many anomalies are infrequent and may defy categorisation; therefore, they are not suited to classification solutions. We test our methods on four discrete datasets (Hospital Sepsis patients, Chess Moves, the Wisconsin Card Sorting Task and BioFamilies) consisting of discrete sequences. Probabilistic Suffix Trees are trained on this data which maintain each discrete symbol's location and position in a given sequence. The trained models are exposed to "new" data where any deviations from learned patterns either in location on the sequence or frequency of occurrence will denote patterns that are unusual compared with the original training data. To assist in the identification of new patterns and to avoid confusing old patterns as new or novel we use Bayesian surprise to detect the discrepancies between what we are expecting and actual results. We can assign the degree of surprise or unexpectedness to any new pattern and provide an indication of why certain patterns are deemed novel or surprising and why others are not.

1 | Introduction

The emotion of surprise is an essential function in many human cognitive and intellectual processes when acquiring new knowledge and skills (Baldi and Itti 2010; Andrew et al. 2013). Surprise is generally described by cognitive scientists as an emotion that occurs when our assumptions and the actual consequences diverge to a greater or lesser extent (Berlyne 1994; Ekman and Davidson 1960). These discrepancies of belief can be assessed by a principled approach using a modification to Bayesian theory which allows us to express our beliefs and to modify these beliefs based on new data input to the system.

We implement a version of the equation devised by Itti and Baldi which models subjective beliefs that are reviewed as new data becomes available (Itti and Baldi 2005). Bayesian surprise can be used as a metric to assess any differences between a model's prior and posterior beliefs. The larger the difference between the two distributions the bigger the surprise metric (Itti and Baldi 2009). *Surprise* as a criterion for judging differences in belief is finding applications in reinforcement-based learning for automating the learning process (Schmidhuber 2010; Gottlieb et al. 2013) and autonomous agents (Rhienberger and Hammitt 2018; Maguire et al. 2019). Furthermore, the creative world of fashion design is starting to realise the benefits of using

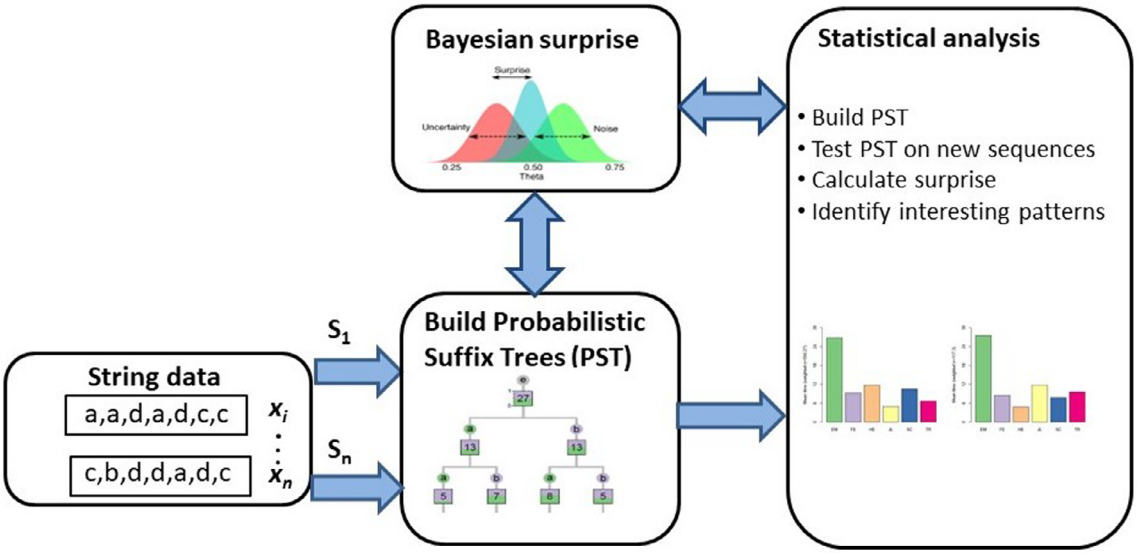

FIGURE 1 | Probabilistic suffix tree generation integrated with Bayesian surprise.

AI for designing consumer products, applying *surprise* as a metric to judge how consumers will perceive unfamiliar product styles and features that may be pleasing and attractive to the eye (Becattini et al. 2017) (Figure 1).

Recently, Bayesian surprise is finding applications in deep learning language models (GPT-2) where event sequences are modelled during storey telling (Kumar et al. 2023). The surprise score is used to measure the listener's change in beliefs when the storey takes an unexpected turn. Generative models such as the GPT family provide a rich source of textual data generation for rich experimentation and analysis (Binz and Schulz 2023). Other recent applications include (Qiao et al. 2022) where Qiao used Bayesian surprise to gain a better understanding of neuron connectivity modulation and brain plasticity. Chieppe et al. (Chieppe et al. 2022) considered Bayesian surprise for the association with good experiences, whilst (Ishikawa et al. 2025) other work used Bayesian surprise to quantify pain with novel, unpleasant experiences (Onysk et al. 2024).

Bayesian surprise is suitable for anomaly detection, which is the process of seeking unusual patterns compared with normal, expected data. Many potentially useful and interesting patterns can be revealed through anomaly detection. The complication in many applications is the infrequency of anomaly occurrence, which may be construed as noise. This often prevents a classification solution, as there may not be enough examples to build a robust model. Additionally, there is no guarantee that new anomalous patterns will have similar characteristics to previously observed trends or patterns.

In the work, we build Probabilistic suffix trees (PST) to represent data sets with variable record sizes of discrete sequences of symbols. In Table 1 we have fictitious data collected from a shop; all possible customer transactions are identified by a letter. The first two transactions are legitimate, with two customers entering; they pick up items and/or put items back on the shelf, then pay for them and then leave. However, transaction 3 is anomalous: the customer picks up several items, places one item back, and leaves the shop with four items but did not pay.

TABLE 1 | Example customer data sequences.

Transaction	String
1.	A, B, C, D
2.	A, B, E, B, B, B, E, C, C, D
3.	A, B, E, B, B, B, B, D

Abbreviations: A = enters shop; B = picks item; C = pays for an item; D = leaves shop; E = put item back on shelf.

The remainder of this is structured as follows: section two considers the related work; section three provides an overview of the theoretical framework; section four provides details of the data and the analytical methods used; section five discusses the results; section six provides the conclusions.

1.1 | Contribution of this Work

In our experiments, we train Probabilistic Suffix Trees (PST) to model the sequences of four symbol-based data sets; these are partitioned into train/test sections. After training, the test data acts as "new" data which is then passed through the PST. PSTs are generative models and provide the probabilities of the expected outputs based on the prior and posterior relationships. The divergence between the two distributions is calculated by the Bayesian Surprise criteria and determines the uniqueness/anomalousness of the new test data. However, we need to distinguish between novel patterns and noise. We consider outliers or noise to be sequences unlikely to have been generated by the model. We can more or less identify outliers by setting a threshold in the prediction quality distribution such that sequences having scores below the threshold will be considered as outliers. The difference between outliers/noise and interesting patterns is explained in the main body text,

but in effect, calculating log-loss, which is a good prediction quality measure between the new data (which in fact could be typical data, or noise, or interesting data) and the probability the PST could have generated the new data. Low probabilities tend to imply the new data record is an outlier; however, this is only an indication.

2 | Related Work and Baseline Methods

The baseline methods commonly used to model discrete sequences are often conducted by analysis of the number and composition of the symbols; also, the length of the sequence and the transition rate from one symbol to another can all provide useful information. Summaries of sequences in terms of what a representative sequence may contain, such as the most frequent sequence and the modal or middle sequence, are provided. Clustering is also used to create homogeneous groups of related sequences; sequences that have different compositions will appear in different clusters. In addition to sequence transition, Shannon entropy is used to measure the diversity of the symbols in any sequence. Distance measures such as the Longest Common Prefix (LCP) and the Longest Common Sub-sequence (LCS) from string theory are used to compute similarities and distances. We explain these in greater detail in the methods section.

Many sequences of symbols often have a hierarchical structure; the SEQUITUR system takes advantage of this characteristic whereby text is composed of letters, sentences and paragraphs. SEQUITUR assembles a data structure from sequences of text symbols. Repeated phrases based on their frequency are replaced with a recursive rule that can reconstruct the sentence or phrase and hence generates the grammar in a hierarchical structure. SEQUITUR simplifies any subsequence that occurs at least once into a rule and performs this operation using recursion. The main advantage is a hierarchical structure that can manage long sequences of symbols; such sequences are usually problematic for many machine learning algorithms (Nevill-Manning and Witten 1997).

Lin and Keogh tackled the conversion of continuous time-series into discrete symbolic components using the Symbolic Aggregate approXimation (SAX) algorithm and the Piecewise Aggregate Approximation (PAA) algorithm (Lin et al. 2007). The PAA algorithm decomposes continuous time series signals into an alphabet of discrete symbols. Their secondary objective was to search for motifs or repeating sub-sequences of symbols; the motifs may represent a sequence of symbols that are naturally grouped together and may represent useful or interesting activity in the time-series (Keogh et al. 2002). The PAA and SAX algorithms were further improved by keeping the information of the continuous time series slope, making it easier for the discretisation of the symbolic representation (Zalewski et al. 2012).

Sequence information is particularly important in Natural Language Processing (NLP) and speech recognition (Rieck and Laskov 2008; Wilson et al. 2007). A major issue in NLP is to avoid ambiguity. Part-of-speech tagging (POS) annotates the sequences of words to help resolve this issue, whereby the position/location of words in a sentence is a major consideration. Often, Hidden Markov Models (HMM) are used to model text data that has been annotated (tag/label) in POS corpora. The HMM advantage over other methods is that word context can be modelled using other words in the near neighbourhood, and they are able to provide probabilities based on the ambiguous word and the previously tagged words based on their location. In order to predict future sequences, strong assumptions are made by Markov chains; the main consideration is that the current state only matters and past states should not influence future predictions. The Markov assumption on the probabilities of any sequence when predicting the future is that the past should not unduly influence the internal states. The characteristics of the HMM make it suitable for many sequential problems (Liao and Fasang 2021; Boldt et al. 2019) especially for anomaly detection in sequences (Florez-Larrahondo et al. 2005).

Recent work by Wang uses HMM for anomaly detection in smart homes, examining behavioural discrete sequences for profiling residents and predicting their actions (Wang et al. 2023). HMM and Probabilistic Suffix Trees (PST) have a biologically plausible mechanism for holding variable length sequences similar to human cognition (Hard et al. 2011). However, Basgol implements a predictive event segmentation model using self-supervised neural networks to achieve similar outcomes (Basgol et al. 2024).

In biology, various string searching algorithms for RNA and DNA sequences have been developed. However, they have a common goal to detect motifs, as they search for recurring sub-sequences in sequential, discrete data (Li and Homer 2010). This leads on to the suffix tree data structure; this is commonly used to hold sequential data and can model words and their location in a sentence. It is a hierarchical data structure that is often used to find the longest sub-string or sub-sequence in a DNA sequence. For example, Huang employed suffix trees to extract periodic patterns from very long temporal sequences and then used self-attention neural transformers (Huang et al. 2021; Huang 2023). Reick conducted experiments on several sequence-based data structures such as tries (a data structure similar to a tree), generalised suffix trees, and data arrays for the analysis of long sequences; this was a very useful analysis comparing and contrasting the strengths of each data structure (Rieck and Laskov 2008). The experiments were conducted on a variety of data sets from bioinformatics, text processing, and cyber-security network intrusion.

3 | Theoretical Framework

3.1 | Probabilistic Suffix Trees

Similar to the HMM, the Probabilistic Suffix Tree is also used for discrete sequence modelling (Largeron-Leténo 2003). Markov processes are suited to detecting anomalies in discrete sequences with the caveat that unusual activity could be represented by an array of chronological observations (Zolfaghari et al. 2021). The Probabilistic Suffix Tree (PST) incorporates a Variable Length Markov Chain (VLMC), with the suffix tree as the basic data structure. In Figure 2 we give an example structure trained on a repeating pattern. Using a VLMC allows variable length sequences (effectively lagging variables)

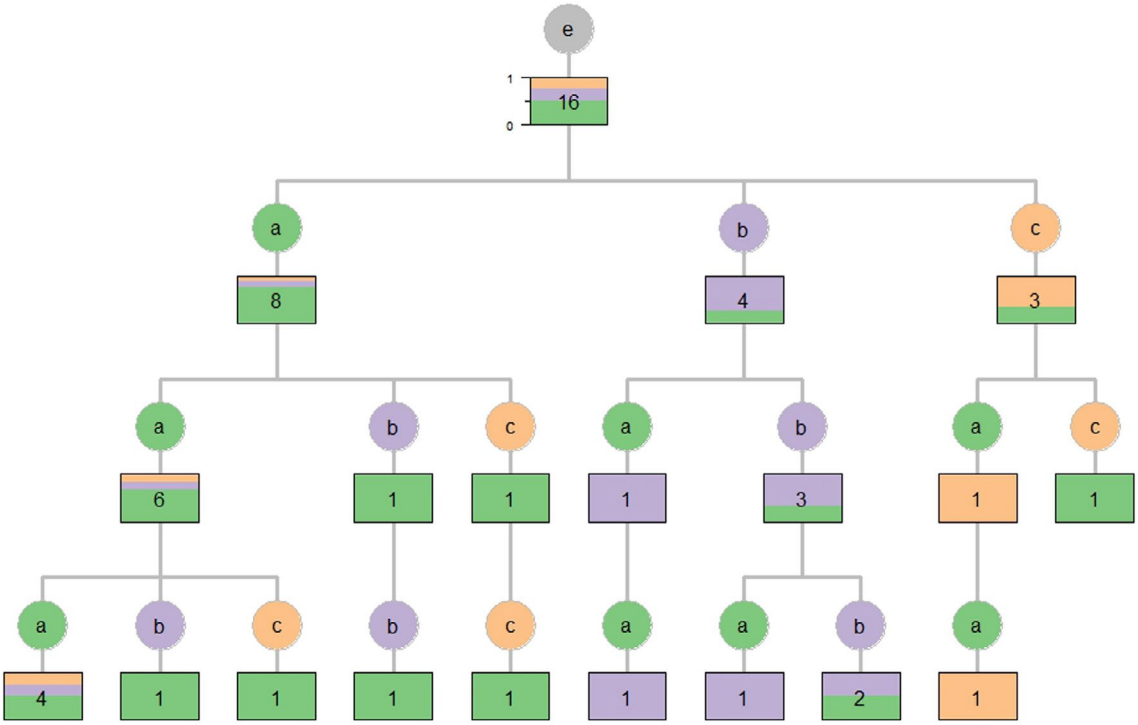

FIGURE 2 | Probabilistic Suffix Tree example, trained on the symbol sequence "c-c-a-a-a-a-b-b-b-b-a-a-a-a-c-c". The tree has a maximum depth of three with a minimum support requirement of one occurrence for a symbol to be incorporated into the tree. The node values are counts of that symbol's appearance. For the root node (e), we have 16 symbols in total, the next layer has 8, 4, and 3 ($n-1$). That is 4 "c's", 8 "a's" and 4 "b's". The frequency count for symbol "c" is $n-1$ since it is the first symbol in the sequence. The probability distribution for each node is shown as a barplot.

to be used in a given data set when training and testing the PST. Otherwise, the data would be constrained by fixed length sequences such as some neural networks require, like Long Short Term Memory, which is a major disadvantage since the data set requires padding (usually with zeros) up to the longest length sequence. Furthermore, the VLMC property enables emission probabilities to be calculated, and thus a predictive model can be constructed (Berchtold 2010). As each state is dependent only on the previous state, probabilities need to be defined for the next state. Having knowledge of the current state, equations for state probabilities and VLMC as implemented from the Traminer software (Gabadinho et al. 2011; Gabadinho and Ritschard 2016):

$$P(x_i|x_{i-1},\ \ldots,x_1) = P(x_i|x_{i-1})$$

The probability of the sequence can be decomposed into:

$$P(x) = P(x_L, x_{L-1},\ \ldots, x_1) = P(x_L|x_{L-1})P(x_{L-1}|x_{L-2})\ldots P(x_2|x_1)P(x_1) \quad (1)$$

$P(x_1)$ can also be calculated from the transition probabilities, multiplying the initial state probabilities at time $t=0$ by the transition matrix, the probabilities of states at time $t=1$ can be derived and therefore we also have them for time $t=n$.

VLMCs model sequential data without recourse to complex estimation procedures but they have significantly better performance compared with HMMs (Bulmann and Wyner 1999). Furthermore, one great advantage is the VLMC generative ability to compute a probability distribution and hence make a prediction on what the next sequences should be, based on the learned sequences. Based on the PST model S developed from training data, we can generate new test sequence likelihoods. The new sequences are passed back into the trained PST which generates the conditional probabilities for the next expected symbols to be predicted. Equation (2) generates these sequences, with the expectation that sequences with low probabilities could be of interest to the user and perhaps anomalous.

$$\forall \ell \in \{0,1,2,\ \ldots\}: \sum_{x \in A^\ell} P^S(x) = 1 \quad (2)$$

where is the alphabet of symbols is defined by A and S is the generative model representing the probability A^ℓ. Whilst $x \in A^\ell$ is the sequence presented to the PST (S). These probabilities are assessed by the Bayesian surprise algorithm to determine if a pattern or sequence is surprising or interesting.

The tree in Figure 2 is constructed from the training sequence "c-c-a-a-a-a-b-b-b-b-a-a-a-a-c-c". The tree has a maximum depth of three with a minimum support requirement of one occurrence for a symbol to be incorporated into the tree. The node values are counts of that symbol's appearance. For the root node (e), we have 16 symbols in total; the next layer has 8, 4, and 3 ($n-1$).

The PST can be constructed from a single sequence of symbols or a series of sequences. Here in our example, we have a single sequence. In Figure 3 the PST is built by successively adding contexts of increasing length k. A node labelled with the context c = c1, ..., ck stores the conditional probability of

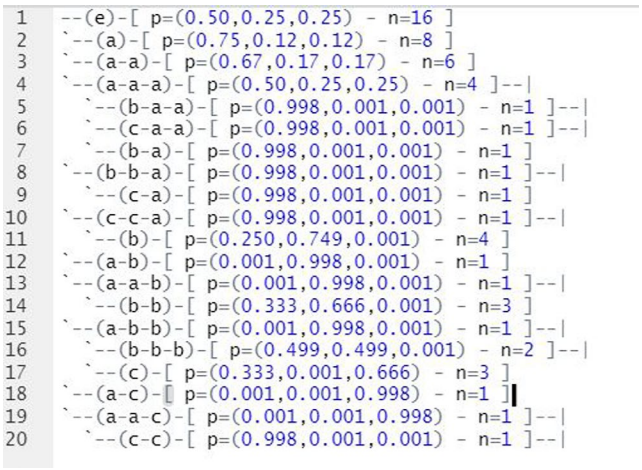

FIGURE 3 | Internal structure and probabilities of the probabilistic suffix tree.

observing the next symbol in the sequence. In line 1, the (e) symbol denotes the root node. The nodes labelled by a, b and c have the root node e as parent (the longest proper suffix3 of a string of length 1 is the empty string e). A leaf node has no child and this occurs when the maximal context length is reached; they are denoted by the "–|" symbol. The four distinct subsequences of length 2 (a-a, a-b, b-a, and a-c) appearing in the training sequence are then added to the tree. We also have subsequences of length 3.

3.2 | Bayesian Surprise and Novelty

The connection between human reasoning and Bayesian modelling is the assumption that Bayesian cognitive theories are effectively a rational analysis grounded on the observations of an initial theory and revising it based on new data (Lee and Wagenmakers 2013). There is sufficient evidence for assuming the Bayesian approach for making models of cognition is essentially correct. However, there are several counterarguments where humans deviate from Bayesian inference (Lee and Wagenmakers 2013; Griffiths and Tenenbaum 2006). The evidence is based on several human problem-solving tasks that produce consistent results when tasks become too difficult to manage using normative techniques and thus become reliant on heuristic approaches (Bain 2016). The usual convention for stating Bayes rule is given below:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{3}$$

where $P(h|D)$ is the posterior probability of the hypothesis h given the data D; $P(D|h)$ is the likelihood of D given h; $P(h)$ is the prior probability of hypothesis h; $P(D)$ is the marginal likelihood of the probability of the data D.

The Bayesian surprise measure S, which tests the two-fold variation between prior and posterior over the hypothesis and data and returns a value (Baldi and Itti 2010; Itti and Baldi 2009). This value will be either positive or negative depending on the observers belief in the hypothesis when it either increases or decreases. The *distance* measure used is the Kullback–Leibler divergence measure. Several applications have recently used the Bayesian Surprise criteria to as part of a feedback criteria for improving the reliability of machine learning models such as neural networks and thematic maps (Hasanbelliu et al. 2012; Correll and Heer 2017; Grassi and Bartels 2021).

$$S(D,h) = distance\big[P(h), P(h|D)\big] \tag{4}$$

The Bayesian surprise measure provides a natural and useful method for defining and representing novel and surprising patterns (Andrew et al. 2013). Equation (5) calculates the distribution over all hypothesis $h \in \mathscr{H}$. The surprise is given as the two-fold difference between $P(h|D)$ and $P(h)$. The model space is defined by M, in this case the output from the probabilistic suffix trees but could be from any generative type model.

$$\begin{aligned} S(D,\mathscr{H}) &= distance\big[P(h), P(h|D)\big] \\ &= \sum_{\mathscr{H}} P(M|D) log \frac{P(M|D)}{P|M} \end{aligned} \tag{5}$$

Novelty and surprise play a fundamental role in human and animal behaviour for survival, attention and adaptation. Surprise, is not however, entirely related to the information content of a pattern alone (Itti and Baldi 2009; Bayarri and Morales 2003). Experiments with patterns of visual white-noise (random but with high information content) presented to participants over time, their Bayesian surprise quickly decreased and soon vanished. This occurred as they adjusted their beliefs so that the random patterns are anticipated and expected. "*Thus, more informative data may not always be more important, interesting, worthy of attention, or surprising*" (Baldi and Itti 2010). Shannon or similar information theoretic measures would erroneously classify the majority of unusual patterns as surprising because of their low probability.

The Kullback–Leibler (KL) divergence examines the relative entropy (Kullback and Leibler 1951) between prior and posterior distributions (Berger et al. 2009). It is defined by:

$$KL\big[p(y) \,\|\, p(x)\big] = \sum_{i=1}^{n} p(y_i) log \frac{p(y_i)}{p(x_i)} \tag{6}$$

where $p(y)$ represents the posterior or correct distribution of data and $p(x)$ represents the hypothesis or model. We obtain a value measuring the difference between the prior distribution $p(\theta)$ to the posterior distribution $p(\theta|y)$ (Statisticat, LLC 2020). The machine learning perspective of a novel pattern is deemed to be a statistical outlier that is different to the probability density function of previously observed patterns (Marsland 2003), in other words novel patterns are those with low estimated probability of occurrence.

However, without some sort of memory or the ability to recognise previous interesting patterns, each presentation of data would result in similar outcomes of interesting scores being assigned. This can be tackled in one of two ways: either incorporate the new data patterns into the PST and retrain it or use a decay function over time that will dampen surprise such as that proposed by Baldi (Baldi and Itti 2010).

$$\frac{1}{a_n} + log\left(1 - \frac{1}{a_N + b_N}\right) \approx \frac{1-p}{p_N} \quad (7)$$

where N is the number of data samples, a and b refer to the respectively to the prior and posterior probability values.

4 | Data and Analytical Methods

All of the data sets consist of strings of symbols, usually letters with one or two numbers (as strings). The data sequences for each record can be variable length or fixed length. The sequences identify a series of discrete actions or conditions which have been recorded; the location and position of each symbol in the sequence may be important. The datasets generally contain other values such as biological measurements, gender, age, etc. represented by numeric factors and continuous values. This data is interesting, but we do not use it; we only use the sequences of events/actions. In Table 2 we summarise the data; the missing data records were removed.

4.1 | Bio-Family Data Set

The Swiss Panel collected data over a 16-year-long period of family life sequences (Mueller et al. 2007). The individuals selected for the study were born between 1909 and 1972 and contain details of 2000 individuals aged from 15 to 30 years. The sequence length is fixed at 16, with some missing data. There are eight states based on single and a combination of family situations defined from the combination of five basic states such as Living with parents (P), Married (M), Divorced (D), Left home (L), Having Children (C). In Table 3 an example of six records for the 11-year sequences is given.

TABLE 2 | Overview of the data sets.

Data	No of records	length of sequence	Unique symbols
BioFam	2000	Fixed (16)	8
Sepsis	1000	Variable (3–30)	16
Chess	280	Variable (2–10)	191
WCST	360	Fixed (60)	18

For the first record with ID = 1335, we see that this individual lived with their parents from 2002 to 2010, then left home in 2011, then from 2012 onwards was married and in 2013 had a child.

4.2 | Sepsis Data Set

Sepsis occurs when the body fights an infection but then causes the antibodies to attack the patient's own cells. It is a serious condition usually requiring hospitalisation and may damage the internal organs such as the liver, kidneys, and lungs. In the worst-case scenario, it may also lead to death. The majority recover from a mild case of sepsis, but for septic shock, the mortality rate is about 30%–40% (NICE 2024). The data consists of cases collected from a Canadian hospital and represents sequential events of medical interventions to combat sepsis. Each record/case represents a patient's pathway through the hospital system. There are approximately 1000 patient cases with about 15,000 interventions (with 16 unique possibilities.) recorded for each patient. Also, a maximum of 39 variables were collected, test results from clinical parameters, and medications prescribed. Each patient can have a variable number of interventions and outcomes, giving a variable length of the discrete sequence records.

The symbol set for all records will contain at least three of the following:

AdmissionC; AdmissionNC; CRP; ERRegistration; ERSepsis; Triage; ERTriage; IVAntibiotics; IVLiquid; LacticAcid; Leucocytes; ReleaseA; ReleaseB; ReleaseC; ReleaseD; ReleaseE; ReturnER.

Where: ER (emergency room); Triage (prioritise patient treatment); Leucocytes (test for white blood cell count); CRP (C-reactive protein, a test for sepsis); IV Antibiotics (Intravenous antibiotics), IV Liquid (Intravenous fluid); Lactic Acid (tests for Lactic Acid which is affected by sepsis).

In Table 4 an example of the variable length sequences is shown; however, we only show the first eight sequences for this dataset. On average, the typical number of symbols in a sequence is 13, the smallest length is three symbols and the longest is 33 symbols. Clearly, the more symbols for a given patient, then the more serious the infection. The patient's entry into the hospital system begins with ERReg and with a discharge of one of the five types (typically Release A), but if a patient relapses, then a return to ER is likely.

TABLE 3 | 16 years of biofamily records listing six individuals, 1st column refers to the record id, each year contains their current state where (P) indicates living with parents, (L) left home, (LM) left home and married, (LMC) left home and married with children.

ID	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2012	2013	2014	2015	2016
1335	P	P	P	P	P	P	P	P	P	L	LM	LM	LMC	LMC	LMC	LMC
1516	P	P	L	L	L	L	L	L	L	L	L	L	L	L	L	L
1870	P	P	P	P	P	P	P	P	P	LM	LM	LM	LM	LM	LM	LM
2162	P	P	P	P	P	P	P	P	P	P	P	LM	LM	LM	LM	LMC
398	P	P	P	P	L	L	L	L	L	LM	LMC	LMC	LMC	LMC	LMC	LMC
902	P	P	P	P	P	P	LMC	LMC	LMC	LMC	LMC	LMC	LMC	LMC	LMC	LMC

TABLE 4 | Six patient records truncated at eight symbols for each.

ID	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
841	ERReg	ERTriage	LacticAcid	Leucocytes	CRP	ERSepsisTriage	IVLiquid	IVAntibiotics
825	ERReg	ERTriage	ERSepsisTriage	IVLiquid	Leucocytes	CRP	LacticAcid	IVAntibiotics
430	ERReg	ERTriage	CRP	Leucocytes	LacticAcid	ERSepsisTriage	IVLiquid	IVAntibiotics
95	ERReg	ERTriage	ERSepsisTriage	CRP	Leucocytes	IVLiquid	IVAntibiotics	AdmissionNC
209	ERReg	ERSepsisTriage	ERTriage	IVLiquid	IVAntibiotics	CRP	Leucocytes	LacticAcid
442	ERReg	ERTriage	ERSepsisTriage	CRP	Leucocytes	LacticAcid	IVLiquid	IVAntibiotics

4.3 | Wisconsin Card Sorting Test (WCST)

The Wisconsin Card Sorting Test (Berg 1948) was designed to reveal cognitive processes such as perseverance, attention, abstract thinking, and set shifting (Lange et al. 2016). It can measure the so-called perseverative behaviours that refer to the participants fixation on incorrect behaviour. The data set used in this work is generated using the PsyToolkit software and contains 30 healthy staff and students aged between 24 and 50 from the University of Sunderland. The test we used is a variation based on the original WCST, which is preferably used on those with cognitive issues. The participants are presented with 60 cards, and for each card, they must select one of four rules they believe the card should belong to. The rules can be colour of object (red, blue, green), shape of object (star, circle, triangle, cross) and the number of objects. The rules change after the presentation of 10 cards, which tests the participants ability to change strategy when presented with an incorrect answer. The software presents the sequence of cards and the participants response with either the correct or incorrect answer. It also provides the total number of errors and the perseveration errors (old strategy) and non-perseveration errors. All participants will make errors since this is a feedback mechanism informing them the old strategy no longer works and they must figure out a new one. In Figure 6 the first 10 sequences (from the same participant) are shown; the test type (number of patterns, pattern type or colour) is the first symbol and remains constant for 10 sequences. The symbols representing the details of the card presented to the participant are next in the sequence; finally, the result is presented, either correct, fail, or out of time.

The experimental software is freely available from the PsyToolkit platform.

https://www.psytoolkit.org/experiment-library/wcst.html (Desrochers et al. 2022).

4.4 | Chess Short Games

The Lichess Chess Game Dataset contains data of 20,058 individual games of chess, extracted from the website lichess.org (Lichess Data Kaggle 2024). The data set contains multiple columns of variables, but of interest to us are the following two properties: moves, i.e., the set of moves played in the game using standard algebraic chess notation [https://www.chess.com/terms/chess-notation]; winner, i.e., the result of the game, which can take one of three values: black, white, or draw. We have also reduced the size of the data set for computational purposes to only include games which are 10 moves or less. This reduces the size of the data set to 280 individual games. An explanation of algebraic chess notation can be found on the chess.com website, but as an example, the moves (e4 e5 Qh5 Nc6 Bc4 Nf6 Qxf7#) for the game below can be described in Table 5. Players (white or black) take turns in sequence, and each move highlights the finishing place of the piece (the starting position can normally be inferred as pieces have standard moves), with special symbols for example, x for when a piece is captured, + when the King is in check, and # for checkmate.

TABLE 5 | Record of one chess game.

Move code	Meaning
e4	White pawn in e2 moves to e4
e5	Black pawn in e7 moves to e5
Qh5	White queen in d4 moves to h5
Nc6	Black knight in b8 moves to c6
Bc4	White bishop in f3 moves to c4
Nf6	Black knight in g8 moves to f6
Qxf7#	White queen in h5 moves to f7 and has checkmate, so white wins the game

Chess involves cognitive abilities, including reasoning and memorisation. In chess, there are a large variety of opening moves which players need to recall and act on. The most well-known opening move in chess is Sicilian Defence, and there are multiple variations of this, such as e4 c5 Nf3 d6 d4 cxd4 Nxd4 Nf6 Nc3. There are several articles which highlight the various cognitive benefits of chess. For example, Sala and Gobet (Sala and Gobet 2016) found that young chess players can make good mathematicians and highlight how it can help in developing their problem-solving and critical skills, whereas other researchers have explored how chess may be a positive factor in protecting the older population against dementia (Lillo-Crespo et al. 2019).

4.5 | Process

Data records with missing values are removed and we divide the training/test split randomly (75/25). The object of this work is not to build a classifier, but to train a model that can identify how different new patterns are compared with the trained Probabilistic Suffix Trees (PST). When new patterns appear, the PST will generate probabilities as to the likelihood these sequences differ from what has been learned. We use RStudio as the programming environment and load the PST package and the TraMineR package developed by Gabadinho (Gabadinho et al. 2011; Gabadinho and Ritschard 2016) to ensure the symbols in the sequence are formatted as required by the PST. As the test data is inputted to the PST for every symbol in every test record, the difference between the probabilities is noted. The 1st record is the probabilities assigned to the training data symbols as deemed by the PST. The 2nd record is the probabilities for the test data as they are passed through the PST.

In Algorithm 1 the input takes a data structure consisting of either fixed length or variable length strings. The algorithm will output a trained PST and the probabilities for each symbol in the training data. The Parameters are used to train the PST and do not require much in the way of tuning. The L parameter is an integer value and sets the maximal depth of the PST. The *nmin* parameter is an integer value and controls the minimum number of occurrences of a string to add it in the tree. The parameter *ymin* is also an integer value and controls the smoothing for conditional probabilities, assuring that no symbol, and hence no sequence, is predicted to have a null probability. The parameter *ymin* sets a lower bound for a symbol's probability.

ALGORITHM 1 | Train Probabilistic Suffix Tree.

Input: set of string data D.
Output: Trained PST $\mathbf{PST}_n$; Probabilities for each symbol $\mathbf{P_{sym}}$.
Parameters: $[L = 10; nmin = 2, ymin = 0.001]$.
1: Convert $Seq \leftarrow D$ using seqdef().
2: Split $Seq_{train}, Seq_{test} \leftarrow Seq$ by 75/25.
3: Train PST using *pstree*([*Parameters*]) on Seq_{train}.
4: Obtain probabilities for each $\mathbf{P_{sym}} \leftarrow$ PST.
5: cprob(PST, L = 0, prob. = TRUE).
6: *Return* $[\mathbf{P_{sym}}; \mathbf{PST_n}]$.

ALGORITHM 2 | Calculate Bayesian Surprise.

Input: Trained PST $\mathbf{PST}_n$; Probabilities for each symbol $\mathbf{P_{sym}}$; Test data $\mathbf{T}_n$.
Output: $\boldsymbol{\beta}_\mathbf{n}$; $\mathbf{TP_n}$; Bayesian Surprise $\mathbf{Surp_n}$; Entropy $\mathbf{E_n}$; KullBack–Leibler $\mathbf{KL_n}$.
1: Initialize $\mathbf{KL_n}; \mathbf{E_n}; \mathbf{Surp_n} = 0$.
2: Initialize $P(Data|\,\theta) = 0$, $P(\theta|\,Data) = 0$, $P(\theta) =$ dbinom(0.5).
3: **repeat**.
4: Obtain the Likelihood $P(Data|\,\theta) = P_{sym_n}$.
5: Calculate the Posterior $P(\theta|\,Data) = P(Data|\,\theta)$ x $P(\theta|\,Data)$.
6: Calculate the standard Prior $P(\theta) = P(Data|\,\theta)$ x $P(\theta)$.
7: Calculate $Surp_n$ = mean(Posterior) - mean(Prior) **Equation 5**.
8: Calculate $KL_n = KL(Prior, Posterior, log_{10})$.
9: Calculate $E_n =$ $shannon.cond.ent(Prior, Posterior, log_{10})$.
10: Calculate $\beta_n = i$ **Equation 7**.
11: **until** $P_{sym} \notin T_n$.
12: **Return** $[\mathbf{Surp_n}; \mathbf{KL_n}; \mathbf{E_n}; \mathbf{TP_n}, \boldsymbol{\beta}_\mathbf{n}]$.

In lines 1–2, the string data is converted into special sequence format (for the PST) and then split 75/25 into train and test partitions. In line 3, the PST is training on this data using the parameters. Lines 4–5 generate the probabilities for each unique symbol in the PST, the *prob* parameter can be set to probabilities or relative frequencies. Line 6 returns the trained PST and the associated symbol probabilities.

In Algorithm 2 the trained PST, Equations (5) and (7) use the test data to generate probabilities. For input the algorithm receives a trained PST, probabilities for each symbol in the PST and the test data. It will output upon completion for each test record: the decay values β_n; the Bayesian Surprise $Surp_n$; Entropy E_n; KullBack–Leibler distance KL_n. In lines 1–2, the values for KL_n, E_n and $Surp_n$ are set to zero, these will be calculated for each and every record in the test data. The Prior and Posterior values are set to zero, the likelihood value is set to a binomial function that is centred at 0.5. This is our main assumption (belief) in our approach, it assumes that overall, any

given pattern has a 0.5 probability of being surprising. Where $P(Data|\theta)$ is the likelihood, $P(\theta|Data)$ is the prior, and $P(\theta)$ is the marginal likelihood.

Lines 4–6, perform the Bayesian inference stage by calculating the likelihood, the Posterior and the standardised Posterior. The posterior must be standardised, this is an important property of any probability density or mass function is that it integrates to one. The likelihood refers to the probability of observing the data that has been observed assuming that the data came from a specific scenario. The posterior can be computed from three key values: 1. A likelihood distribution, P(Data|θ) 2. A prior distribution, P(θ) 3. The average likelihood.

Lines 7–10 calculate the Bayesian surprise $Surp_n$; the Shannon Entropy E_n; KullBack–Leibler distance KL_n and decay values β_n. These values are stored in vectors for later use for comparisons. Using decay, how many patterns are genuinely interesting prior to reaching the zero value cut-off? This is a process based on similar patterns reappearing over time, as they are presented. Line 11, reiterates the loop until all of the test patterns have been analysed. Line 12 returns the results for identification of those patterns deemed to be interesting/novel.

After calculating the entropy, KL distance, and Bayesian Surprise for the three datasets, we determine if a test pattern is surprising or not. However, it is probably more important to determine why a given sequence is surprising. This is based on the use of a number of measures to analyse the structure, composition, and regularity of the interesting sequences. We have six methods that assess the sequences:

1. Sequence entropy. Shannon entropy is used to measure the diversity of the states or symbols in any sequence, based on the length of the sequence and the number of symbols in the alphabet (Oliveira and Ospina 2018). A more varied sequence will have a higher entropy than a sequence composed of fewer symbols.
2. Sequence complexity. A sequence may be defined in terms of the complexity of distinct sub-sequences that can be discovered from the distinct state sequences and is often called turbulence in the literature (Elzinga 2010).
3. Longest Common Prefix (LCP). Distance measures from string theory are used to compute similarities and distances. The longest common prefix for the sequences is the common prefix between the two most dissimilar strings (Elzinga and Studer 2015).
4. Longest Common Sub-sequence (LCS). A sub-sequence is a relaxation of the idea of a sub-string; a sub-sequence is a pattern that appears in the same relative order but is not necessarily contiguous (Ritschard 2021). This method is particularly well suited to DNA symbol matching (Needleman and Wunsch 1970). The computations produce a matrix that can be clustered for further information. The individual records, once clustered, provide an indication of their similarity.
5. Sequence event transitions. Rather than simply examining the sequences of symbols, we can also observe the sequences of transitions or events between sequences deemed as surprising and those that are not. The actual transitions between symbols might reveal why they are of interest.

5 | Results

We now compare the values of entropy, Kullback–Leibler (KL) distance and Bayesian surprise for each test data set. All three measures are based on the differences between prior and posterior probabilities. It should be noted that the line plots have a more or less regular appearance. This is because similar patterns occur in the test data; the smaller the number of unique symbols, the more likely the symbol sequences will be similar and hence repeated over time. The "Wow" level is the two-fold difference of the Bayesian Surprise value when derived from the differences between the prior and posterior values as devised by Itti (Itti and Baldi 2009). This is calculated for each dataset and will be unique in each case. It should be noted that the Bayesian Surprise is a much more conservative measure, especially when coupled with the "Wow" threshold which requires a two-fold increase of Bayesian Surprise for any pattern in the test data to be considered interesting. Entropy, in all three datasets, has a much higher response to the differences between prior and posterior values, i.e., the overall information gain is high. The KL distance is more conservative than entropy but still can be extremely variable across the four datasets.

Viewing the Biofam data results, shown in Figure 8, the information value of the KL measure fluctuates between a range of 0.3–1.2 and thus has high information content. Similarly, the entropy has high information content but does not fluctuate to any great extent. The Bayesian Surprise measure for Biofam is just over the zero value; the "Wow" level is also just above zero. Shown in Figure 4, the regularity of the Biofam data is evident; 10 records are shown, each with 16 sequences. The majority of the individuals (younger people) in the study are still living with parents, but in many sequences we can observe life changes over the period of 16 years as the individuals marry, have children, move from home, and parents, etc.

Examining the Sepsis data results, shown in Figure 9, the information value of the KL measure fluctuates between a range of 0.0–0.5. However, in this data entropy has a much higher information content than the other measures and fluctuates between 0.75 and 1.00. The Bayesian Surprise measure for Sepsis is between 0.1 and 0.35 and has similar characteristics to the KL measure. The "Wow" level is around 0.23. Shown in Figure 5, the varied size of the sequences comprising the Sepsis data is evident; however, the sequences always start with "ERRegistration" and usually terminate with "ReleaseA".

Viewing the Chess data results, shown in Figure 10, the information value of the KL measure fluctuates between a range of 0.0–0.9. Entropy has a much higher information content than the other measures, and fluctuates between 0.90 and 1.00. The Bayesian Surprise measure for Chess is between 0.0 and 0.25. The "Wow" level is around 0.1. Shown in Figure 7, the size of the sequences varies between 2 and 1; however 0, the sequences usually start with "e4" or "e5" and as the game progresses, more symbols are used to describe the moves.

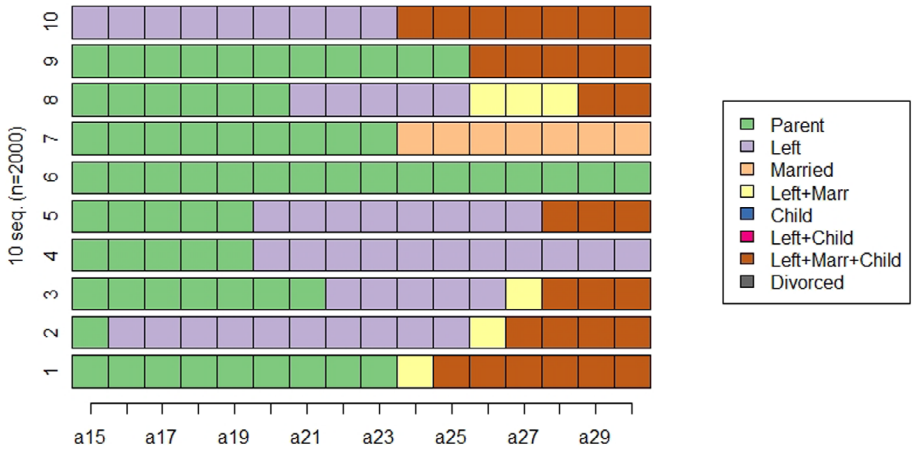

FIGURE 4 | Sequence ordering of biofam symbols for first 10 records, where (P) indicates living with parents, (L) left home, (LM) left home and married, (LMC) left home and married with children, (D) divorced.

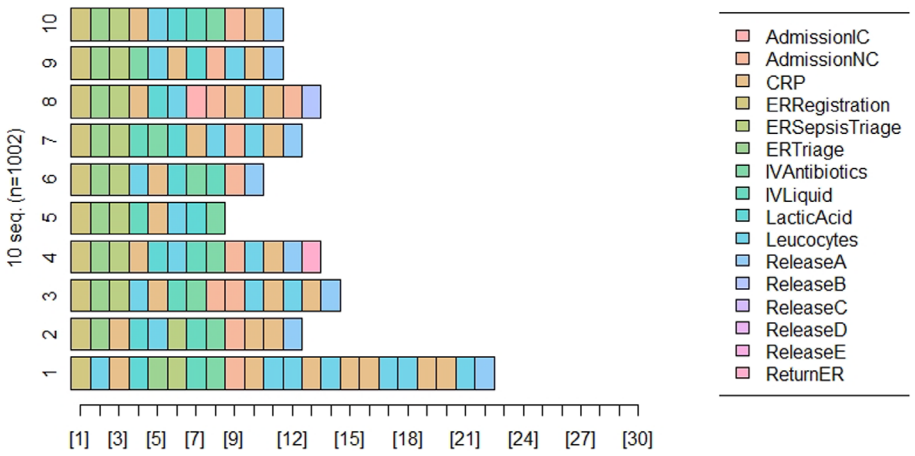

FIGURE 5 | Sequence ordering of sepsis symbols for first 10 records. Main point of interest is the variable sequence length; the first sequence is 11 symbols long and the last is 22 symbols. The symbols are grouped into admission types for emergency room/hospital, clinical test groups, and release from emergency room/hospital groups.

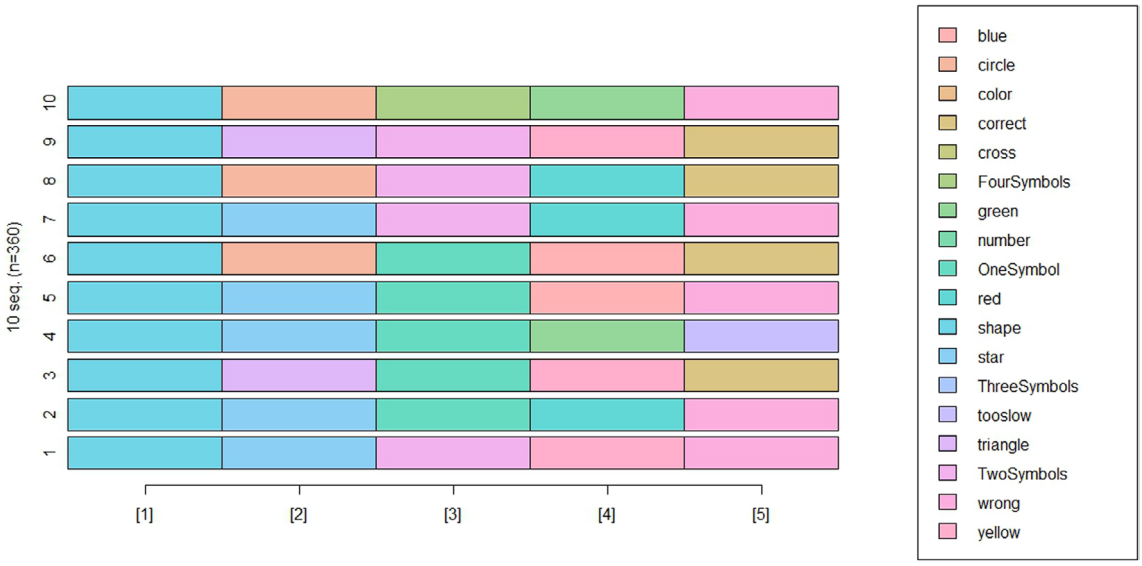

FIGURE 6 | Sequence ordering of Wisconsin card sorting test symbols for first 10 sequences. The first symbol in the sequence is always the test type (shape, colour, number) which changes after 10 symbols, then the shape, colour and number of the selected card, last symbol is always the test result (wrong, correct or too slow). There are 18 possible symbols, as shown in the legend.

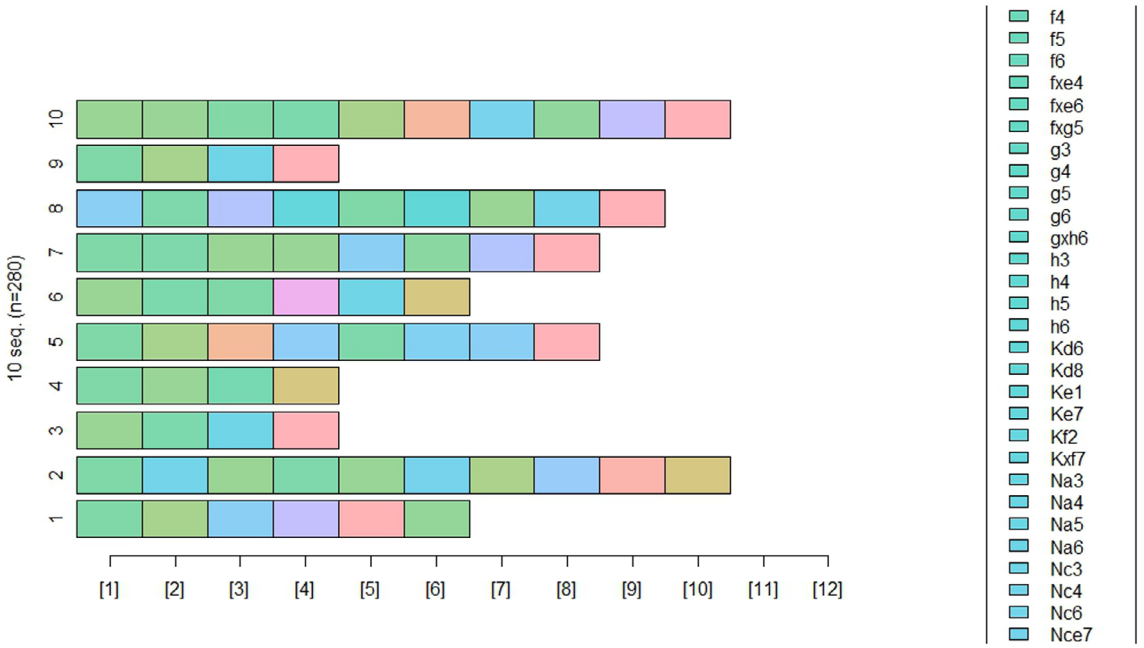

FIGURE 7 | Sequence ordering of first ten chess games of a total of 280, the sequences vary in length from 2 to 10 symbols (game moves), average game length is 6 moves. The legend shows the repertoire of different moves made in these 280 games.

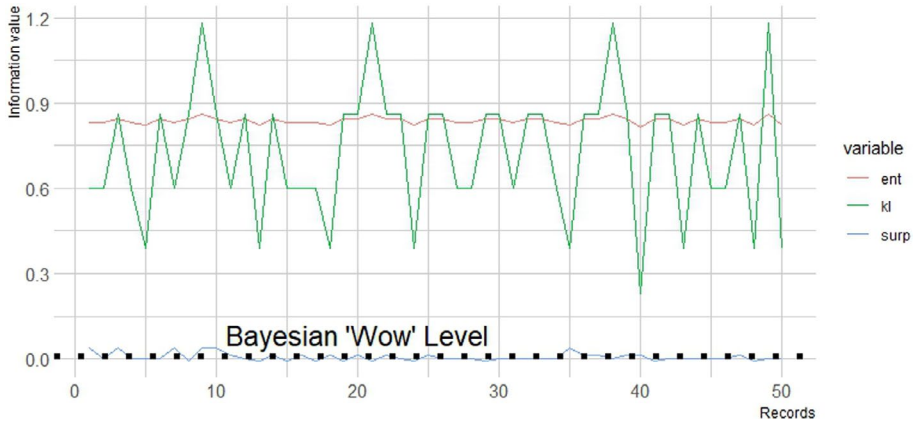

FIGURE 8 | KL, entropy and Bayesian surprise on Biofam data for first 50 records without interest decay. Where: *ent* is the entropy, *KL* is the Kullback-Leibler measure and *surp* is the Bayesian surprise value.

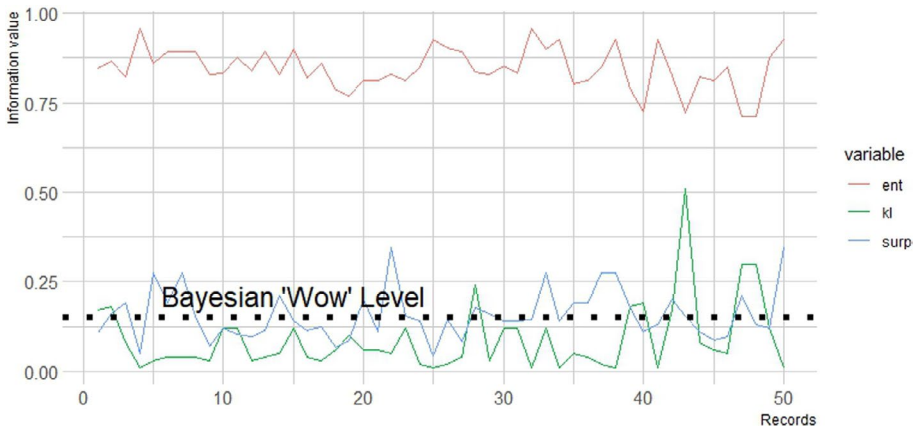

FIGURE 9 | KL, entropy and Bayesian surprise on the Sepsis medical data for first 50 records without interest decay. Where: *ent* is the entropy, *KL* is the Kullback Leibler measure and *surp* is the Bayesian surprise value.

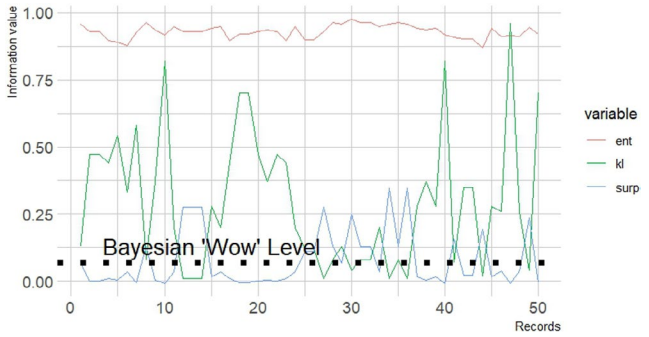

FIGURE 10 | KL, entropy and Bayesian surprise on CHESS for first 50 records without interest decay. Where: *ent* is the entropy, *KL* is the Kullback– Leibler measure and *surp* is the Bayesian surprise value.

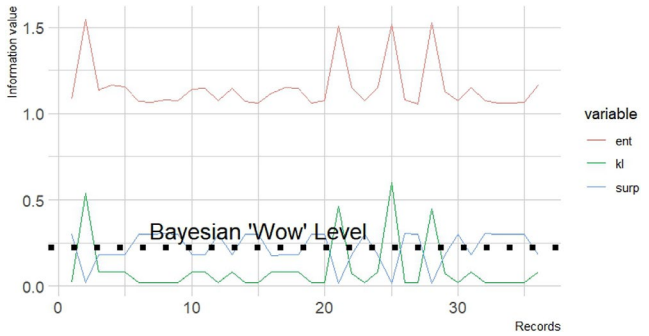

FIGURE 11 | KL, entropy and Bayesian surprise on WCST for first 50 records without interest decay. Where: *ent* is the entropy, *KL* is the Kullback Leibler measure and *surp* is the Bayesian surprise value.

Examining the WCST data results, shown in Figure 11, the information value of the KL measure fluctuates between a range of 0.0–0.6. Entropy fluctuates between 1.00 and 1.50. The Bayesian Surprise measure for Sepsis is between 0.1 and 0.35. The "Wow" level is around 0.25. Shown in Figure 6, the regular structure of the sequences comprising the WCST data with its equal sizes of five symbols and the first symbol is repeated 10 times, before it changes to another symbol for 10 repeats and so forth. This predicable regularity makes for a less "interesting" data set information-wise.

5.1 | Estimates of Noise and Errors

Next we must determine the amount of noisy data present in each dataset; we provide a summary of the log-loss errors in Table 6 for all datasets. This process is dependent on the quality and quantity of the data used to train the PSTs and should only be used as a rough guide and is not exact. The PSTs are generative models, and we use this feature to see if the test data could have been generated by the PST model. In Figures 12 and 13 we have the log-loss error for an individual Biofam record and for the entire Biofam dataset respectively. The histograms are generated by passing the test data into the trained Biofam PST tree and observing the probabilities of the outputs. The probabilities based on the log-loss error provide an indication of the amount of error overall and individually for the data. The average log-loss error is used as a cut-off point to determine if those particular test records are noise/outliers based on discrepancies between them and the features learned by the PST model.

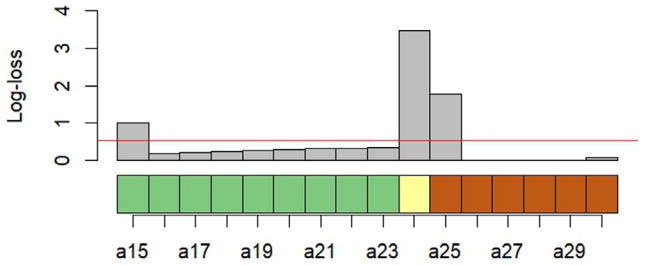

FIGURE 12 | Biofam log-loss error for a single record, the red line indicates the average log-loss and the lower bars are the symbols for that record. The coloured bars are assigned automatically for each unique symbol in the sequence.

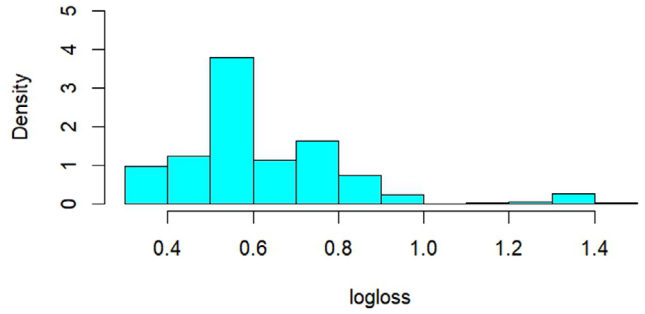

FIGURE 13 | Biofam log-loss error density distribution over the entire data set.

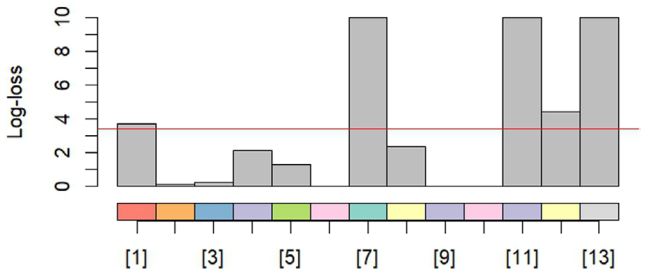

FIGURE 14 | Sepsis log-loss error for a single record, the red line indicates the average log-loss and the lower bars are the symbols for that record. The coloured bars are assigned automatically for each unique symbol in the sequence.

In Figures 14 and 15 histograms for the Sepsis data are presented. The log-loss errors are slightly higher than those of the other datasets, potentially a result of a varied set of sequences with up to 16 symbols and a sequence length ranging between 3 and 30 symbols. Although we have 1000 samples, it is likely that more data is needed to span the input space (the curse of dimensionality).

In Figures 16 and 17 the histograms for the Chess data are displayed. Although the chess data consists of short variable length sequences (2–10) it has a larger than usual repertoire of symbols (191). Again, this will contribute to the log-loss error. The sequence shown in Figure 16 consists of five symbols (five moves)

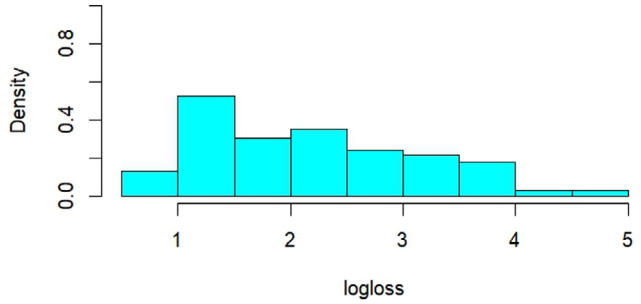

FIGURE 15 | Sepsis log-loss error density distribution over the entire data set.

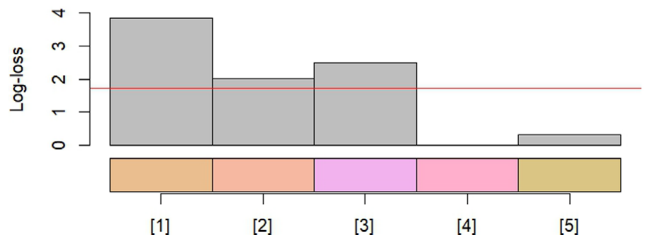

FIGURE 16 | Chess log-loss error for a single record, the red line indicates the average log-loss and the lower bars are the symbols for that record. The coloured bars are assigned automatically for each unique symbol in the sequence.

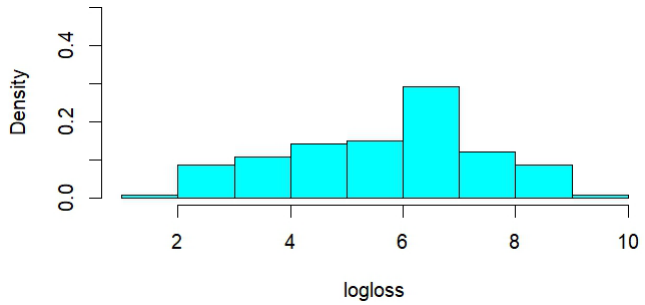

FIGURE 17 | Chess log-loss error density distribution over the entire data set.

which is about the average size for this dataset and is representative of the whole.

In Figures 18 and 19 the histograms for the WCST are shown. This data set has a very regular structure of 60 symbols for each participant with feedback symbols for the answer (correct, wrong or out of time). It has moderate log-loss error per sample/record and the average is similar to the other datasets, although the histogram is somewhat skewed. The interesting aspect about this data set is that it may be structured in different ways, sequences of length 60 or sequences of length 10 (six for each participant).

5.1.1 | Evaluation of Results—Statistics

The next stage is to examine the details of the sequences. We wish to determine if any differences exist between surprising and non-surprising patterns. Referring to Table 6 the results for entropy, sequence complexity, longest common prefix, and longest common sub-sequence are presented.

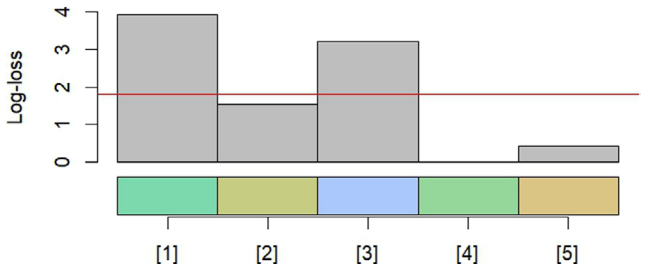

FIGURE 18 | WCST log-loss error for a single record, the red line indicates the average log-loss and the lower bars are the symbols for that record. The coloured bars are assigned automatically for each unique symbol in the sequence.

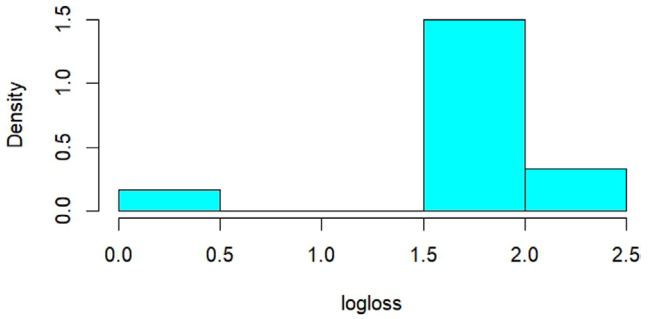

FIGURE 19 | WCST log-loss error density distribution over the entire data set.

For the Sepsis sequences, the average entropy is 0.74, much larger than Biofam (0.35) indicating a varied set of symbols. The turbulence or complexity of the Sepsis data is 11.9 and is more complex than Biofam (4.8) data (which is to be expected) given the symbol set. The longest common prefix for the Sepsis data is 20, which is smaller than Biofam (32). The longest common sub-sequence for Sepsis data is 10; this is smaller than what could be expected given the long lengths of some sequences. Biofam is much larger at 30; this is unusual. Bayesian surprise value is 0.15 for the Sepsis data and is larger than the other three, which is to be expected from the wider range of symbols and variable length size of the sequences.

The Chess sequences form an average entropy of 0.37 with a complexity (Turbulence) rating of 5.3 suggesting, with the values of the other measures of LCP, LCS, Bayes and log-loss of 5.66, that this data set is problematic. As suggested in the other experiments, it is the large number of potential symbols that is likely the cause of the high error rate and identifying a number of records as noise.

Examining the WCST sequences we find that the log-loss is moderate, suggesting there are few noisy patterns in this data set. Entropy and complexity are moderate, along with the length LCP/LCS subsequences discovered.

5.1.2 | Evaluation of Results—Sequence Event Transitions

For all data sets, we tag the sub-sequences with the initial "surprising" and "not-surprising" labels from the test data as it is passed through the Probabilistic Suffix Tree. The sub-sequence events correspond to potentially interesting or not interesting

pattens which are shown in Table 7 we highlight the statistically significant events from sub-sequences that discriminate between interesting and not-interesting sub-sequences for the BioFam data. Usually, we are interested in detecting frequently occurring transitions with event sequences. The first column provides the sequence ID, we show five sequences. In the second column the full sub-sequence is shown, creates a distinct (from-state > to-state) event for every discovered transition consisting of a pair of events (end-state event, start-state event) which is assigned to each transition. Similar to association rules these state to state events have a support value based on a minimum support of the number of similar sequences.

Examining the biofam event sub-sequences further, we discover the most discriminating sequences between surprising and not surprising. The chi-square test is used to test between the two types of sequences; it gives a *p*-value for significance and the Pearson's coefficient.

In Table 8 the statistically significant events from the Sepsis data sub-sequences that discriminate between interesting and not-interesting sub-sequences are presented, using the tagged sequences with the surprising label. The Surprising patterns have values for the key variables between 0.6 and 0.9, whilst the not-surprising sequences events have values much lower between 0.05 and 0.5—only the Lecoucyte > CRP sequence (ID = 1) in Table 8 has a significantly higher value.

The Chess subsequences shown in Table 9 show predominantly the opening moves that are ordered by the support statistic for each sequence. The first row with ID =1 highlights the transition between Nf3 to Nc6; the white player always moves first. We can deduce they are moving their knight to the f3 square, and the black player is moving their knight to square c6. The 2nd row shows e4-Nf3-Nc6; the subsequence algorithm has picked up the Double King's Pawn Games and the Double King's Pawn Opening. The *surprising statistic* and *p*-value indicate these subsequence moves are valid.

The table indicates the 10 most discriminating sequences for identifying surprising and not surprising patterns. The magnitudes of the values for the surprising patterns are much larger than the not-surprising patterns, and this serves to differentiate between them.

In Table 10 the key sequences for the WCST are shown. The *p*-values are not significant; however, similar to the Chess data, the values of the surprising versus the not-surprising subsequences are very different (0.23 versus 0.07) and thus can discriminate between them.

Our work just uses the discrete symbol sequences to detect unusual and novel patterns. Although the data sets have additional information such as participant demographics and other variables, we do not use these. We took the decision to concentrate only on sequences as our main objective. Furthermore, data sets with variable length sequences are often problematic for many probabilistic machine learning methods; however, the PST is well suited to this task. The results have shown that variable length sequences, like the Sepsis data which has several different symbols, are the most interesting to analyse. Such sequences allow a richer diversity of patterns to be generated and can capture interesting patterns occurring in the

TABLE 6 | Anomaly detection results on the data sets.

Dataset	Entropy	Complexity	LCP	LCS	Bayes surp	Ave log-loss density
BioFam	0.35	4.8	32	30	0.009	0.61
Sepsis	0.74	11.9	20	10	0.15	2.18
CHESS	0.37	5.3	20	8	0.098	5.66
WCST	0.66	4.1	15.0	5	0.18	1.68

TABLE 7 | Event sub-sequences discriminate between surprising and not-surprising sequences for biofam data.

ID	subseq)	Sup	*p*	Statistic	Surprising	Not-surprising
1	(Parent)-(Parent > Left)	0.43	0.00	119.53	0.32	0.57
2	(Parent > Left)	0.43	0.00	119.53	0.32	0.57
3	(Parent)-(Parent > Married)	0.12	0.00	37.81	0.16	0.07
4	(Parent > Married)	0.12	0.00	37.81	0.16	0.07
5	(Left > Left + Marr)	0.23	0.00	32.72	0.18	0.29
6	(Parent)-(Left > Left + Marr)	0.23	0.00	32.13	0.18	0.29
7	(Parent)-(Parent > Left)-(Left > Left + Marr)	0.23	0.00	32.13	0.18	0.29
8	(Parent > Left)-(Left > Left + Marr)	0.23	0.00	32.13	0.18	0.29
9	(Parent)-(Parent > Left + Marr)	0.25	0.00	15.80	0.29	0.21
10	(Parent > Left + Marr)	0.25	0.00	15.80	0.29	0.21

TABLE 8 | Event sub-sequences discriminate between surprising and not surprising sequences for Sepsis data.

ID	subseq	Support	*p*	Statistic	Surp	Not-surp
1	(Leucocytes > CRP)-(CRP > Leucocytes)	0.39	0.00	75.48	0.61	0.06
2	(Leucocytes > CRP)-(Leucocytes > CRP)	0.35	0.00	67.87	0.56	0.05
3	(ERRegistration > ERTriage)-(Leucocytes > CRP)-(CRP > Leucocytes)	0.37	0.00	67.86	0.58	0.06
4	(ERRegistration)-(Leucocytes > CRP)-(CRP > Leucocytes)	0.36	0.00	66.39	0.57	0.06
5	(ERRegistration > ERTriage)-(Leucocytes > CRP)-(Leucocytes > CRP)	0.33	0.00	63.75	0.53	0.04
6	(ERRegistration)-(ERRegistration > ERTriage)-(Leucocytes > CRP)	0.35	0.00	62.11	0.55	0.06
7	(ERRegistration)-(Leucocytes > CRP)-(Leucocytes > CRP)	0.33	0.00	60.82	0.53	0.05
8	(ERRegistration)-(ERRegistration > ERTriage)-(Leucocytes > CRP)	0.32	0.00	59.67	0.51	0.04
9	(Leucocytes > CRP)	0.73	0.00	58.26	0.91	0.47
10	(CRP > Leucocytes)-(Leucocytes > CRP)	0.37	0.00	58.07	0.57	0.09

TABLE 9 | Event sub-sequences discriminate between surprising and not surprising sequences for CHESS data.

ID	subsequence	Support	*p*	Statistic	Surprising	Not-surprising
1	(Nf3 > Nc6)	0.14	0.01	14.08	0.23	0.00
2	(e4)-(Nf3 > Nc6)	0.12	0.02	12.20	0.21	0.00
3	(e4)-(e4 > e5)-(e5 > Nf3)-(Nf3 > Nc6)	0.09	0.22	7.76	0.15	0.00
4	(e4)-(e4 > e5)-(Nf3 > Nc6)	0.09	0.22	7.76	0.15	0.00
5	(e4)-(e5 > Nf3)-(Nf3 > Nc6)	0.09	0.22	7.76	0.15	0.00
6	(e4 > e5)-(e5 > Nf3)-(Nf3 > Nc6)	0.09	0.22	7.76	0.15	0.00
7	(e4 > e5)-(Nf3 > Nc6)	0.09	0.22	7.76	0.15	0.00
8	(e5 > Nf3)-(Nf3 > Nc6)	0.09	0.22	7.76	0.15	0.00
9	(Nf3 > Nc6)-(white > %)	0.08	0.33	6.92	0.14	0.00
10	(e4)-(Nf3 > Nc6)-(white > %)	0.07	0.47	6.09	0.12	0.00

TABLE 10 | Event sub-sequences discriminate between surprising and not surprising sequences for WCST data.

ID	subsequence	Support	*p*	Statistic	Surprising	Not-surprising
1	(shape)-(shape > star)	0.19	0.71	3.16	0.30	0.00
2	(shape > star)	0.19	0.71	3.16	0.30	0.00
3	(shape)-(shape > star)-(star > OneSymbol)	0.14	0.97	1.72	0.22	0.00
4	(shape)-(star > OneSymbol)	0.14	0.97	1.72	0.22	0.00
5	(shape > star)-(star > OneSymbol)	0.14	0.97	1.72	0.22	0.00
6	(star > OneSymbol)	0.14	0.97	1.72	0.22	0.00
7	(number)-(blue > correct)	0.14	1.00	0.49	0.09	0.23
8	(red > correct)	0.22	1.00	0.26	0.17	0.31
9	(shape)	0.36	1.00	0.02	0.39	0.31
10	(colour)	0.33	1.00	0.02	0.30	0.38

data. In fact, data sets with a limited set of symbols and with fixed length sequences generally do not produce a lot of interesting patterns.

After training, the PST is presented with new data and will output probability scores for each sequence. These new scores are in effect the posterior distributions to be compared with the prior distributions for each and every symbol generated by the trained PST. The differences between prior and posterior estimates are compared using the Bayesian Surprise via the KL measure; this determines how interesting or anomalous these new patterns are. A further concept that must be taken into account is the decay parameter. If this was not considered, there would be no "memory" or history of previously learned sequences, and the PST would repeatedly consider all new sequences with scores above the "Wow" cut-off point as unusual or novel. Any assessment of pattern novelty has to be external to whatever model is used; therefore, we did not incorporate the new data by retraining the PST.

6 | Discussion and Conclusions

This work advances knowledge for the detection of unusual discrete sequence data and provides some explanation of why a pattern can be considered unusual or interesting. We have examined why sequence data can be considered surprising using criteria such as sequence composition and complexity, entropy measures, and state transitions from one symbol to the next in a sequence. Many outlier/anomalous detection methods rank patterns based on infrequence; Shannon's theory is the usual way to assess them based on their low probability. However, simply using low probability scores is suboptimal for identifying interesting patterns, as all such patterns would be regarded as interesting. Bayesian surprise is not so misled, unlike information theoretic measures such as Shannon surprise and entropy, for example. This work, therefore, presents a plausible, cognitive-inspired framework for detecting unusual sequences and by reducing the interest signal when we encounter similar patterns. The surprising patterns identified by our methods should be helpful to the data analyst, but there is a degree of subjectivity as to what constitutes an interesting pattern. We can say our system is internally self-consistent, based on the history and similarity of past sequences. Our future work will use Neural Networks such as Recurrent Networks and Long-Short Term Memory (LSTM) as these methods can manage longer sequences through their enhanced memory.

Author Contributions

Ken McGarry: conceptualisation, methodology, software, experimentation and writing. **David Nelson:** software, data collection and writing.

Acknowledgements

We would like to thank the anonymous reviewers for their advice for improving this work and Satu Helske for advice on the variable length, hidden markov model approach. We would also acknowledge the use of the PsyToolkit developed by Professor Gijsbert Stoet.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are openly available in github at https://github.com/kenmcgarry/BayesSurprise.

References

Andrew, B., M. Marco, and B. Gianluca. 2013. "Novelty or Surprise?" *Frontiers in Psychology* 4: 907.

Bain, R. 2016. "Are Our Brains Bayesian?" *Significance* 13, no. 4: 14–19.

Baldi, P., and L. Itti. 2010. "Of Bits and Wows: A Bayesian Theory of Surprise with Applications to Attention." *Neural Networks* 23: 649–666.

Basgol, H., A. Inci, and U. Emre. 2024. "Predictive Event Segmentation and Representation with Neural Networks: A Self-Supervised Model Assessed by Psychological Experiments." *Cognitive Systems Research* 83: 101167.

Bayarri, M. J., and J. Morales. 2003. "Bayesian Measures of Surprise for Outlier Detection." *Journal of Statistical Planning and Inference* 111: 3–22.

Becattini, N., Y. Borgianni, G. Cascini, and F. Rotini. 2017. "Surprise and Design Creativity: Investigating the Drivers of Unexpectedness." *International Journal of Design Creativity and Innovation* 5, no. 1–2: 29–47.

Berchtold, A. 2010. "Sequence Analysis and Transition Models." In *Encyclopedia of Animal Behavior*, edited by M. Breed and J. Moore, 139–145. Academic Press.

Berg, E. 1948. "A Simple Objective Technique for Measuring Flexibility in Thinking." *Journal of General Psychology* 39, no. 1: 15–22.

Berger, J., J. Bernardo, and D. Sun. 2009. "The Formal Definition of Reference Priors." *Annals of Statistics* 37, no. 2: 905–938.

Berlyne, D. 1994. *Conflict, Arousal, and Curiosity*. Oxford University Press.

Binz, M., and E. Schulz. 2023. "Using Cognitive Psychology to Understand GPT3." *Proceedings of the National Academy of Sciences of the United States of America* 120, no. 6: e2218523120.

Boldt, M., A. Borg, S. Ickin, and J. Gustafsson. 2019. "Anomaly Detection of Event Sequences Using Multiple Temporal Resolutions and Markov Chains." *Knowledge and Information Systems* 62: 669–686.

Bulmann, P., and A. Wyner. 1999. "Variable Length Markov Chains." *Annals of Statistics* 27, no. 3: 480–513.

Chieppe, P., P. Sweetser, and E. Newman. 2022. "Bayesian Modelling of the Well-Made Surprise." In *Proceedings of the 13th International Conference on Computational Creativity*, edited by M. Hedblom, vol. 126, 135. ICCC.

Correll, M., and J. Heer. 2017. "Surprise! Bayesian Weighting for De-Biasing Thematic Maps." *IEEE Transactions on Visualization and Computer Graphics* 23, no. 1: 651–660.

Desrochers, T. M., A. Aarit, M. R. Maechler, S. Jorja, Y. R. Nadira, and M. E. Berryhill. 2022. "Caught in the ACTS: Defining Abstract Cognitive Task Sequences as an Independent Process." *Journal of Cognitive Neuroscience* 34, no. 7: 1103–1113.

Ekman, P., and R. Davidson. 1960. *The Nature of Emotion: Fundamental Questions*. McGraw-Hill.

Elzinga, C. 2010. "Complexity of Categorical Time Series." *Sociological Methods & Research* 38, no. 3: 463–481.

Elzinga, C., and M. Studer. 2015. "Spell Sequences, State Proximities, and Distance Metrics." *Sociological Methods & Research* 44, no. 1: 3–47.

Florez-Larrahondo, G., S. Bridges, and R. Vaughn. 2005. "Efficient Modeling of Discrete Events for Anomaly Detection Using Hidden Markov Models." *Information Security* 3650: 506–514.

Gabadinho, A., and G. Ritschard. 2016. "Analyzing State Sequences with Probabilistic Suffix Trees: The PST R Package." *Journal of Statistical Software* 72, no. 3: 1–39.

Gabadinho, A., G. Ritschard, N. Müller, and M. Studer. 2011. "Analyzing and Visualizing State Sequences in R with TraMineR." *Journal of Statistical Software* 40, no. 4: 1–37.

Gottlieb, J., P. Oudeyer, M. Lopes, and A. Baranes. 2013. "Information-Seeking, Curiosity, and Attention: Computational and Neural Mechanisms." *Trends in Cognitive Sciences* 17: 585–593.

Grassi, P., and A. Bartels. 2021. "Magic, Bayes and Wows: A Bayesian Account of Magic Tricks." *Neuroscience and Biobehavioral Reviews* 126: 515–527.

Griffiths, T. L., and J. Tenenbaum. 2006. "Optimal Predictions in Everyday Cognition." *Psychological Science* 17, no. 9: 767–773.

Hard, B., G. Recchia, and B. Tversky. 2011. "The Shape of Action." *Journal of Experimental Psychology: General* 140, no. 4: 586–604.

Hasanbelliu, E., K. Kampa, J. Príncipe, and J. T. Cobb. 2012. "Online Learning Using a Bayesian Surprise Metric." *2012 International Joint Conference on Neural Networks (IJCNN)*: 1–8.

Huang, J. 2023. "Self-Attention-Based Long Temporal Sequence Modeling Method for Temporal Action Detection." *Neurocomputing* 554: 126617.

Huang, J., B. Jaysawal, and C. Wang. 2021. "Mining Full, Inner and Tail Periodic Patterns with Perfect, Imperfect and Asynchronous Periodicity Simultaneously." *Data Mining and Knowledge Discovery* 35: 1225–1257.

Ishikawa, R., G. Ono, and J. Izawa. 2025. "Bayesian Surprise Intensifies Pain in a Novel Visual-Noxious Association." *Cognition* 257: 106064.

Itti, L., and P. Baldi. 2005. A Principled Approach to Detecting Surprising Events in Video.

Itti, L., and P. Baldi. 2009. "Bayesian Surprise Attracts Human Attention." *Vision Research* 49, no. 10: 1295–1306.

Keogh, E., S. Lonardi, and B. Chiu. 2002. *Finding Surprising Patterns in a Time Series Database in Linear Time and Space*, 550–556. Association for Computing Machinery.

Kullback, S., and R. Leibler. 1951. "On Information and Sufficiency." *Annals of Mathematical Statistics* 22, no. 1: 79–86.

Kumar, M., A. Goldstein, S. Michelmann, J. Zacks, U. Hasson, and K. Norman. 2023. "Bayesian Surprise Predicts Human Event Segmentation in Story Listening." *Cognitive Science* 47, no. 10: e13343.

Lange, F., B. Kröger, A. Steinke, C. Seer, R. Dengler, and B. Kopp. 2016. "Decomposing Card-Sorting performance: Effects of Working Memory Load and Age-Related Changes." *Neuropsychology* 30, no. 5: 579–590.

Largeron-Leténo, C. 2003. "Prediction Suffix Trees for Supervised Classification of Sequences." *Pattern Recognition Letters* 24, no. 16: 3153–3164.

Lee, M., and E. Wagenmakers. 2013. *Bayesian Cognitive Modeling*. Cambridge University Press.

Li, H., and N. Homer. 2010. "A Survey of Sequence Alignment Algorithms for Next-Generation Sequencing." *Briefings in Bioinformatics* 11, no. 5: 473–483.

Liao, T., and A. Fasang. 2021. "Comparing Groups of Life-Course Sequences Using the Bayesian Information Criterion and the Likelihood-Ratio Test." *Sociological Methodology* 51, no. 1: 44–85.

Lichess Data Kaggle. 2024. Accessed February 27, 2024. https://www.kaggle.com/datasets/datasnaek/chess.

Lillo-Crespo, M., M. Forner-Ruiz, J. Riquelme-Galindo, D. Ruiz-Fernández, and S. García-Sanjuan. 2019. "Chess Practice as a Protective Factor in Dementia." *International Journal of Environmental Research and Public Health* 16, no. 12: 2116.

Lin, J., E. Keogh, L. Wei, and S. Lonardi. 2007. "Experiencing SAX: A Novel Symbolic Representation of Time Series." *Data Mining and Knowledge Discovery* 15, no. 2: 107–144.

Maguire, P., P. Moser, R. Maguire, and M. Keane. 2019. "Seeing Patterns in Randomness: A Computational Model of Surprise." *Topics in Cognitive Science* 11, no. 1: 103–118.

Marsland, S. 2003. "Novelty Detection in Learning Systems." *Neural Computing Surveys* 3: 1–39.

Mueller, N., M. Studer, and G. Ritschard. 2007. Classification de parcours de vie à l'aide de l'optimal matching.

Needleman, S., and C. Wunsch. 1970. "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins." *Journal of Molecular Biology* 48: 443–453.

Nevill-Manning, C., and I. Witten. 1997. "Identifying Hierarchical Structure in Sequences: A Linear-Time Algorithm." *Journal of Artificial Intelligence Research* 7, no. 1: 67–82.

NICE. 2024. "What is the Prognosis of Sepsis." Accessed January 10, 2024. https://cks.nice.org.uk/topics/sepsis/background-information/prognosis/.

Oliveira, H., and R. Ospina. 2018. A Note on the Shannon Entropy of Short Sequences.

Onysk, J., N. Gregory, M. Whitefield, et al. 2024. "Statistical Learning Shapes Pain Perception and Prediction Independently of External Cues." *eLife* 12: 12.

Qiao, L., L. Zhang, and A. Chen. 2022. "Brain Connectivity Modulation by Bayesian Surprise in Relation to Control Demand Drives Cognitive Flexibility via Control Engagement." *Cerebral Cortex* 34, no. 5: 1985–2000.

Rhienberger, C., and J. Hammitt. 2018. "Dinner with Bayes: On the Revision of Risk Beliefs." *Journal of Risk and Uncertainty* 57, no. 3: 253–280.

Rieck, K., and P. Laskov. 2008. "Linear-Time Computation of Similarity Measures for Sequential Data." *Journal of Machine Learning Research* 9: 23–48.

Ritschard, G. 2021. *Measuring the Nature of Individual Sequences*. Sociological Methods & Research.

Sala, G., and F. Gobet. 2016. "Do the Benefits of Chess Instruction Transfer to Academic and Cognitive Skills." *Educational Research Review* 18: 46–57.

Schmidhuber, J. 2010. "Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990–2010)." *IEEE Transactions on Autonomous Mental Development* 2, no. 3: 230–247.

Statisticat, LLC. 2020. "LaplacesDemon: Complete Environment for Bayesian Inference," R package version 16.1.4.

Wang, X., J. Liu, S. J. Moore, C. D. Nugent, and Y. Xu. 2023. "A Behavioural Hierarchical Analysis Framework in a Smart Home: Integrating HMM and Probabilistic Model Checking." *Information Fusion* 95: 275–292.

Wilson, W., P. Birkin, and U. Aickelin. 2007. "The Motif Tracking Algorithm." *International Journal of Automation and Computing* 5, no. 1: 32–44.

Zalewski, W., F. Silva, F. Wu, H. Lee, and A. Maletzke. 2012. "A Symbolic Representation Method to Preserve the Characteristic Slope of Time Series." In *SBIA'12*, 132–141. Springer-Verlag.

Zolfaghari, S., E. Khodabandehloo, and D. Riboni. 2021. "TraMiner: Vision-Based Analysis of Locomotion Traces for Cognitive Assessment in Smart-Home." *Cognitive Computation* 14, no. 5: 1549–1570.