DEEP NEURAL NETWORKS FOR THE SEQUENTIAL PROBABILITY RATIO TEST ON NON-I.I.D. DATA SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Classifying sequential data as early and as accurately as possible is a challenging yet critical problem, especially when a sampling cost is high. One algorithm that achieves this goal is the sequential probability ratio test (SPRT), which is known as Bayes-optimal: it can keep the expected number of data samples as small as possible, given the desired error upper-bound. The SPRT has recently been found to be the best model that explains the activities of the neurons in the primate parietal cortex that are thought to mediate our complex decision-making processes. However, the original SPRT makes two critical assumptions that limit its application in real-world scenarios: (i) samples are independently and identically distributed, and (ii) the likelihood of the data being derived from each class can be calculated precisely. Here, we propose the SPRT-TANDEM, a deep neural network-based SPRT algorithm that overcomes the above two obstacles. The SPRT-TANDEM estimates the log-likelihood ratio of two alternative hypotheses by leveraging a novel Loss function for Log-Likelihood Ratio estimation (LLLR) while allowing for correlations up to $N (\in \mathbb{N})$ preceding samples. In tests on one original and two public video databases, Nosaic MNIST, UCF101, and SiW, the SPRT-TANDEM achieves statistically significantly better classification accuracy than other baseline classifiers, with a smaller number of data samples. The code and Nosaic MNIST are publicly available at `https://anonymous.4open.science/r/8e802b42-ec6f-4545-b34e-fb320cba4c4d/#home`.

1 INTRODUCTION

The sequential probability ratio test, or SPRT, was originally invented by Abraham Wald, and an equivalent approach was also independently developed and used by Alan Turing in the 1940s (Good, 1979; Simpson, 2010; Wald, 1945). SPRT calculates the log-likelihood ratio (LLR) of two competing hypotheses and updates the LLR every time a new sample is acquired until the LLR reaches one of the two thresholds for alternative hypotheses (Figure 1). Wald and his colleagues proved that when sequential data are sampled from independently and identically distributed (i.i.d.) data, SPRT can minimize the required number of samples to achieve the desired upper-bounds of false positive and false negative rates comparably to the Neyman-Pearson test, known as the most powerful likelihood test (Wald & Wolfowitz, 1948) (see also Theorem (A.5) in Appendix A). Note that Wald used the i.i.d. assumption only for ensuring a finite decision time (i.e., LLR reaches a threshold within finite steps) and for facilitating LLR calculation: the non-i.i.d. property does not affect other aspects of the SPRT including the error upper bounds (Wald, 1947). More recently, Tartakovsky et al. verified that the non-i.i.d. SPRT is optimal or at least asymptotically optimal as the sample size increases (Tartakovsky et al., 2014), opening the possibility of potential applications of the SPRT to non-i.i.d. data series. About 70 years after Wald's invention, neuroscientists found that neurons in the part of the primate brain called the lateral intraparietal cortex (LIP) showed neural activities reminiscent of the SPRT (Kira et al., 2015); when a monkey sequentially collects random pieces of evidence to make a binary choice, LIP neurons show activities proportional to the LLR. Importantly, the time of the decision can be predicted from when the neural activity reaches a fixed threshold, the same as the SPRT's decision rule. Thus, the SPRT, the optimal sequential decision strategy, was re-discovered to be an algorithm explaining primate brains' computing strategy. It remains an open question, however, what algorithm will be used in the brain when the sequential evidence is correlated, non-i.i.d. series.

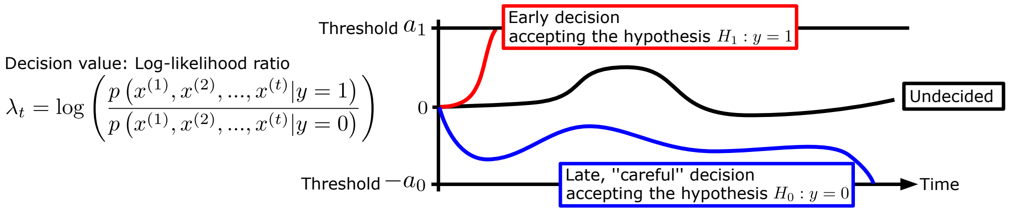
Decision value: Log-likelihood ratio
$\lambda_t = \log\left(\frac{p\left(x^{(1)}, x^{(2)}, ..., x^{(t)} | y=1\right)}{p\left(x^{(1)}, x^{(2)}, ..., x^{(t)} | y=0\right)}\right)$
Threshold a_1
0
Threshold $-a_0$
Early decision
accepting the hypothesis $H_1 : y = 1$
Undecided
Late, "careful" decision
accepting the hypothesis $H_0 : y = 0$
Time

Figure 1: Conceptual figure explaining the SPRT. The SPRT calculates the log-likelihood ratio (LLR) of two competing hypotheses and updates the LLR every time a new sample ($x^{(t)}$) is acquired, until the LLR reaches one of the two thresholds. For data that is easy to be classified, the SPRT outputs an answer after taking a few samples, whereas for difficult data, the SPRT takes in numerous samples in order to make a "careful" decision. For formal definitions and the optimality in early classification of time series, see Appendix A.

The SPRT is now used for several engineering applications (Cabri et al., 2018; Chen et al., 2017; Kulldorff et al., 2011). However, its i.i.d. assumption is too crude for it to be applied to other real-world scenarios, including time-series classification, where data are highly correlated, and key dynamic features for classification often extend across more than one data point, violating the i.i.d. assumption. Moreover, the LLR of alternative hypotheses needs to be calculated as precisely as possible, which is infeasible in many practical applications.

In this paper, we overcome the above difficulties by using an SPRT-based algorithm that Treats data series As an N-th orDEr Markov process (SPRT-TANDEM), aided by a sequential probability density ratio estimation based on deep neural networks. Additionally, we propose a novel Loss function for Log-Likelihood Ratio estimation (LLLR), in order to efficiently estimate the density ratio. The SPRT-TANDEM can classify non-i.i.d. data series with user-defined model complexity by changing $N (\in \mathbb{N})$, the order of approximation, to define the number of past samples on which the given sample depends. By dynamically changing the number of samples used for classification, the SPRT-TANDEM can maintain high classification accuracy while minimizing the sample size as much as possible. Moreover, the SPRT-TANDEM enables a user to flexibly control the speed-accuracy tradeoff without additional training, making it applicable to various practical applications.

We test the SPRT-TANDEM on our new database, Nosaic MNIST (NMNIST), in addition to the publicly available UCF101 action recognition database (Soomro et al., 2012) and Spoofing in the Wild (SiW) database (Liu et al., 2018). Two-way analysis of variance (ANOVA, Fisher (1925)) followed by a Tukey-Kramer multi-comparison test (Tukey, 1949) shows that our proposed SPRT-TANDEM provides statistically significantly higher accuracy than other fixed-length and variable-length classifiers at a smaller number of data samples, making Wald's SPRT applicable even to non-i.i.d. data series. Our contribution is fivefold:

1. We invented a deep neural network-based algorithm, SPRT-TANDEM, that extends Wald's SPRT to non-i.i.d. data series without prior knowledge of the LLR.
2. The SPRT-TANDEM sequentially estimates LLR for earlier and more precise classification of non-i.i.d. data series than other methods, as demonstrated on the three public databases.
3. Using the SPRT-TANDEM, a user can control the speed-accuracy tradeoff and handle variable-length data without additional training.
4. We present the novel loss function, LLLR, to train neural networks for the LLR estimation.
5. We introduce Nosaic MNIST, a novel early-classification database.

2 Related work

The SPRT-TANDEM has multiple interdisciplinary intersections with other fields of research: Wald's classical SPRT, probability density estimation, neurophysiological decision making, and time-series classification. The comprehensive review is left to Appendix B, while in the following, we introduce the SPRT, probability density estimation algorithms, and early classification of the time series.

Sequential Probability Ratio Test (SPRT). The SPRT, denoted by δ^*, is defined as the tuple of a decision rule and a stopping rule (Tartakovsky et al., 2014; Wald, 1947):

Definition 2.1. Sequential Probability Ratio Test (SPRT). *Let λ_t as the LLR at time t. Given the absolute values of lower and upper decision threshold, $a_0 \geq 0$ and $a_1 \geq 0$, SPRT, δ^*, is defined as*

$$\delta^* = (d^*, \tau^*), \tag{1}$$

where the decision rule d^* and stopping time τ^* are

$$d^*(X^{(1,T)}) = \begin{cases} 1 & \text{if } \lambda_{\tau^*} \geq a_1 \\ 0 & \text{if } \lambda_{\tau^*} \leq -a_0 \,, \end{cases} \tag{2}$$

$$\tau^* = \inf\{T \geq 0 | \lambda_T \notin (-a_0, a_1)\} \,. \tag{3}$$

We leave detailed definitions to Appendix A, while an intuitive explanation can be found in Figure 1.

Probability density ratio estimation. Instead of estimating numerator and denominator of a density ratio separately, the probability density ratio estimation algorithms estimate the ratio as a whole, reducing the degree of freedom for more precise estimation (Sugiyama et al., 2010; 2012). Two of the probability density ratio estimation algorithms that closely related to our work are the probabilistic classification (Bickel et al., 2007; Cheng & Chu, 2004; Qin, 1998) and ratio matching (Kanamori et al., 2009; Sugiyama et al., 2008; Tsuboi et al., 2009) algorithms. As we show in Section 3 and 4, the SPRT-TANDEM sequentially estimates the LLR by combining the two algorithms.

Early classification of time series. To make decision time as short as possible, algorithms for early classification of time series can handle variable length of data (Mori et al., 2018; Mori et al., 2016; Xing et al., 2009; 2012) to minimize high sampling costs (e.g., medical diagnostics (Evans et al., 2015; Griffin & Moorman, 2001), or stock crisis identification (Ghalwash et al., 2014)). Leveraging deep neural networks is no exception in the early classification of time series (Dennis et al., 2018; Suzuki et al., 2018). Long short-term memory (LSTM)-s/LSTM-m impose monotonicity on classification score and inter-class margin, respectively, to speed up action detection (Ma et al., 2016). Early and Adaptive Recurrent Label ESTimator (EARLIEST) combines reinforcement learning and a recurrent neural network to decide when to classify and assign a class label (Hartvigsen et al., 2019).

3 Proposed algorithm: SPRT-TANDEM

In this section, we propose the TANDEM formula, which provides the N-th order approximation of the LLR with respect to posterior probabilities. The i.i.d. assumption of Wald's SPRT greatly simplifies the LLR calculation at the expense of the precise temporal relationship between data samples. On the other hand, incorporating a long correlation among multiple data may improve the LLR estimation; however, calculating too long a correlation may potentially be detrimental in the following cases. First, if a class signature is significantly shorter than the correlation length in consideration, uninformative data samples are included in calculating LLR, resulting in a late or wrong decision. Second, long correlations requires calculating a long-range of backpropagation, prone to gradient vanishing problem during training a neural network. Thus, we relax the i.i.d. assumption by keeping only up to the N-th order correlation to calculate the LLR.

The TANDEM formula. Here, we introduce the TANDEM formula, which computes the approximated LLR, the decision value of the SPRT-TANDEM algorithm. The data series is approximated as an N-th order Markov process. For the complete derivation of the 0th (i.i.d.), 1st, and N-th order TANDEM formula, see Appendix C. Given a maximum timestamp $T \in \mathbb{N}$, let $X^{(1,T)}$ and y be a sequential data $X^{(1,T)} := \{x^{(t)}\}_{t=1}^T$ and a class label $y \in \{1, 0\}$, respectively, where $x^{(t)} \in \mathbb{R}^{d_x}$ and $d_x \in \mathbb{N}$. By using Bayes' rule with the N-th order Markov assumption, the joint LLR of data at a timestamp t is written as follows:

$$\begin{aligned} &\log \left(\frac{p(x^{(1)}, x^{(2)}, ..., x^{(t)} | y=1)}{p(x^{(1)}, x^{(2)}, ..., x^{(t)} | y=0)} \right) \\ = &\sum_{s=N+1}^{t} \log \left(\frac{p(y=1|x^{(s-N)}, ..., x^{(s)})}{p(y=0|x^{(s-N)}, ..., x^{(s)})} \right) - \sum_{s=N+2}^{t} \log \left(\frac{p(y=1|x^{(s-N)}, ..., x^{(s-1)})}{p(y=0|x^{(s-N)}, ..., x^{(s-1)})} \right) \\ &- \log \left(\frac{p(y=1)}{p(y=0)} \right) \end{aligned} \tag{4}$$

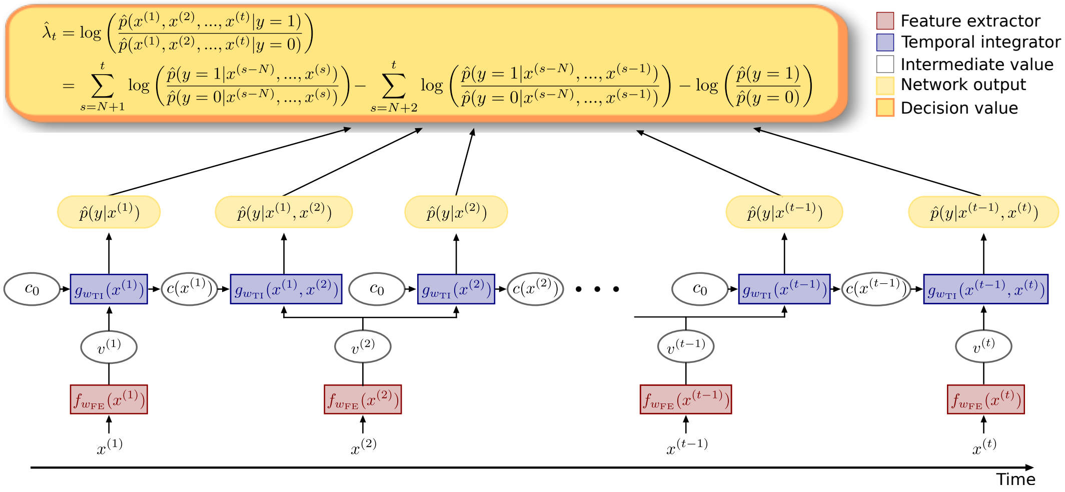
$\hat{\lambda}_t = \log\left(\frac{\hat{p}(x^{(1)}, x^{(2)}, ..., x^{(t)}|y=1)}{\hat{p}(x^{(1)}, x^{(2)}, ..., x^{(t)}|y=0)}\right)$
$= \sum_{s=N+1}^{t} \log\left(\frac{\hat{p}(y=1|x^{(s-N)}, ..., x^{(s)})}{\hat{p}(y=0|x^{(s-N)}, ..., x^{(s)})}\right) - \sum_{s=N+2}^{t} \log\left(\frac{\hat{p}(y=1|x^{(s-N)}, ..., x^{(s-1)})}{\hat{p}(y=0|x^{(s-N)}, ..., x^{(s-1)})}\right) - \log\left(\frac{\hat{p}(y=1)}{\hat{p}(y=0)}\right)$
Feature extractor
Temporal integrator
Intermediate value
Network output
Decision value
$\hat{p}(y|x^{(1)})$
$\hat{p}(y|x^{(1)}, x^{(2)})$
$\hat{p}(y|x^{(2)})$
$\hat{p}(y|x^{(t-1)})$
$\hat{p}(y|x^{(t-1)}, x^{(t)})$
c_0
$g_{w_{\mathrm{TI}}}(x^{(1)})$
$c(x^{(1)})$
$g_{w_{\mathrm{TI}}}(x^{(1)}, x^{(2)})$
c_0
$g_{w_{\mathrm{TI}}}(x^{(2)})$
$c(x^{(2)})$
c_0
$g_{w_{\mathrm{TI}}}(x^{(t-1)})$
$c(x^{(t-1)})$
$g_{w_{\mathrm{TI}}}(x^{(t-1)}, x^{(t)})$
$v^{(1)}$
$v^{(2)}$
$v^{(t-1)}$
$v^{(t)}$
$f_{w_{\mathrm{FE}}}(x^{(1)})$
$f_{w_{\mathrm{FE}}}(x^{(2)})$
$f_{w_{\mathrm{FE}}}(x^{(t-1)})$
$f_{w_{\mathrm{FE}}}(x^{(t)})$
$x^{(1)}$
$x^{(2)}$
$x^{(t-1)}$
$x^{(t)}$
Time

Figure 2: Conceptual diagram of neural network for the SPRT-TANDEM where the order of approximation $N = 1$. The feature extractor (red) extracts the feature vector for classification and outputs it to the temporal integrator (blue). Note that the temporal integrator memorizes up to N preceding states in order to calculate the TANDEM formula (Equation (4)). LLR is calculated using the estimated probability densities that are output from the temporal integrator. We use $\hat{}$ to highlight a quantity estimated by a neural network. Trainable weight parameters are shared across the boxes with the same color in the figure.

(see Equation (82) and (83) in Appendix C for the full formula). Hereafter we use terms k*-let* or *multiplet* to indicate the posterior probabilities, $p(y|x^{(1)}, ..., x^{(k)}) = p(y|X^{(1,k)})$ that consider correlation across k data points. The first two terms of the TANDEM formula (Equation (4)), $N+1$-let and N-let, have the opposite signs working in "tandem" adjusting each other to compute the LLR. The third term is a prior (bias) term. In the experiment, we assume a flat prior or zero bias term, but a user may impose a non-flat prior to handling the biased distribution of a dataset. The TANDEM formula can be interpreted as a realization of the probability matching approach of the probability density estimation, under an N-th order Markov assumption of data series.

Neural network that calculates the SPRT-TANDEM formula. The SPRT-TANDEM is designed to explicitly calculate the N-th order TANDEM formula to realize density ratio estimation, which is the critical difference between our SPRT-TANDEM network and other architecture based on convolutional neural networks (CNNs) and recurrent neural networks (RNN). Figure 2 illustrates a conceptual diagram explaining a generalized neural network structure, in accordance with the 1st-order TANDEM formula for simplicity. The network consists of a feature extractor and a temporal integrator (highlighted by red and blue boxes, respectively). They are arbitrary networks that a user can choose depending on classification problems or available computational resources. The feature extractor and temporal integrator are separately trained because we find that this achieves better performance than the end-to-end approach (also see Appendix D). The feature extractor outputs single-frame features (e.g., outputs from a global average pooling layer), which are the input vectors of the temporal integrator. The output vectors from the temporal integrator are transformed with a fully-connected layer into two-dimensional logits, which are then input to the softmax layer to obtain posterior probabilities. They are used to compute the LLR to run the SPRT (Equation (2)). Note that during the training phase of the feature extractor, the global average pooling layer is followed by a fully-connected layer for binary classification.

How to choose the hyperparameter N? By tuning the hyperparameter N, a user can efficiently boost the model performance depending on databases; in Section 5, we change N to visualize the model performance as a function of N. Here, we provide two ways to choose N. One is to choose N based on the "*specific time scale,*" a concept introduced in Appendix D, where we describe in detail how to guess on the best N depending on databases. The other is to use a hyperparameter tuning algorithm such as Optuna Akiba et al. (2019) to objectively choose N. Note that tuning N is not computationally expensive, because N is only related to the temporal integrator, not to the feature extractor. In fact, the training speed of the temporal integrator is $60 - 200$ times faster than the feature extractor in our experiment.

4 Proposed loss function: multiplet cross-entropy loss and LLLR

Given a maximum timestamp $T \in \mathbb{N}$ and dataset size $M \in \mathbb{N}$, let $S := \{(X^{(1,T)}, y_i)\}_{i=1}^{M}$ be a sequential dataset. Training our network to calculate the TANDEM formula involves the following loss functions in combination: (i) multiplet cross-entropy loss, $L_{\mathrm{multiplet}}$, and (ii) the Loss for Log Likelihood Ratio estimation (LLLR), L_{LLR}. The total loss, L_{total} is defined as

$$L_{\mathrm{total}} = L_{\mathrm{multiplet}} + L_{\mathrm{LLR}} \, . \tag{5}$$

Multiplet cross-entropy loss. Under the N-th order TANDEM formula, correlations among up to $N+1$ data point are taken into account when estimating the LLR. To precisely estimate LLR at any timestamp, we apply a binary cross-entropy loss to all of the multiplets, from singlet to $N+1$-let:

$$L_{\mathrm{multiplet}} := \sum_{k=1}^{N+1} L_{k\text{-let}} \, , \tag{6}$$

where

$$L_{k\text{-let}} := \frac{1}{M(T-N)} \sum_{i=1}^{M} \sum_{t=k}^{T-(N+1-k)} \Big(-\log \hat{p}(y_i | x_i^{(t-k+1)}, ..., x_i^{(t)}) \Big) \, . \tag{7}$$

We use $\hat{p}$ to highlight a probability density estimated by a neural network. Minimizing the multiplet cross-entropy losses is equivalent to minimizing the Kullback-Leibler divergence (Kullback & Leibler, 1951) of the estimated posterior k-let, $\hat{p}(y_i | x_i^{(t-k+1)}, ..., x_i^{(t)})$, and the true posterior $p(y_i | x_i^{(t-k+1)}, ..., x_i^{(t)})$, as we provide a proof in Appendix E.

Loss for Log-Likelihood Ratio estimation (LLLR). As stated above, optimization using the multiplet cross-entropy loss should lead to the precise estimation of true posterior, and consequently, provide the true LLR of data approximated as an N-th order Markov process. However, the optimization process is not guaranteed to reach global minima in an actual experiment. Thus, we propose a novel loss function that optimizes the estimated LLR as a whole, unlike the multiplet cross-entropy loss, which optimizes the TANDEM formula's local components.

To minimize the Kullback-Leibler divergence between the estimated and the true LLRs, we introduce LLLR below:

$$L_{\mathrm{LLR}} = \frac{1}{MT} \sum_{i=1}^{M} \sum_{t=1}^{T} \left| y_i - \sigma \left(\log \left(\frac{\hat{p}(x_i^{(1)}, x_i^{(2)}, ..., x_i^{(t)} | y=1)}{\hat{p}(x_i^{(1)}, x_i^{(2)}, ..., x_i^{(t)} | y=0)} \right) \right) \right| \, , \tag{8}$$

where σ is the sigmoid function. The LLLR can be interpreted as a ratio matching approach of the probability density ratio estimation. As we provide a proof in Appendix F, the LLLR is a normalized variant of KLIEP (Khan et al., 2019; Sugiyama et al., 2008), a ratio matching algorithm minimizing the Kullback-Leibler divergence between the estimated and true density ratio. The original KLIEP algorithm has unbounded terms and often causes a diverging loss value when used as a loss function. In contrast, the LLLR can readily be used together with conventional loss functions such as the cross-entropy loss. In Appendix F, we experimentally test the above theoretical predictions: The LLLR used together with the multiplet cross-entropy loss has a statistically significantly smaller classification error than the loss without LLLR (p-value < 0.001) and the bounded KLIEP loss with the multiplet cross-entropy loss (p-value < 0.001).

5 Experiments and results

In the following experiments, we use two quantities as evaluation criteria: (i) balanced accuracy, the arithmetic mean of the true positive and true negative rates, and (ii) mean hitting time, the average number of data samples used for classification. Note that the balanced accuracy is robust to class imbalance (Luque et al., 2019), and is equal to accuracy on balanced datasets.

Evaluated public databases are NMNIST, UCF, and SiW. Training, validation, and test datasets are split and fixed throughout the experiment. We selected three early-classification models (LSTM-s (Ma et al., 2016), LSTM-m (Ma et al., 2016), and EARLIEST (Hartvigsen et al., 2019)) and one

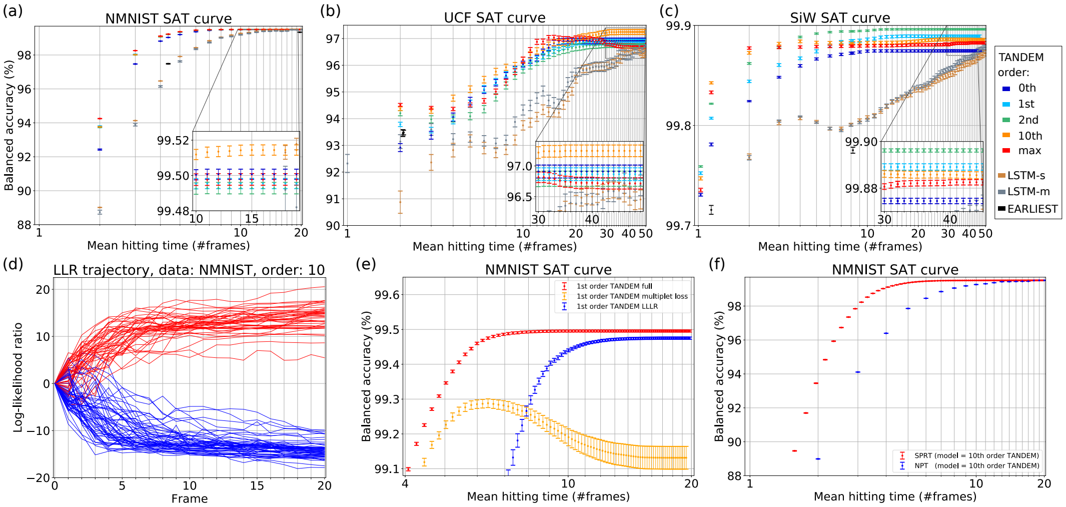

Figure 3: Experimental results. (a-c) Speed-accuracy tradeoff (SAT) curves for three databases: NMNIST, UCF, and SiW. Note that only representative results are shown. Error bars show the standard error of the mean (SEM). (d) Example LLR trajectories calculated on the NMNIST database with the 10th-order SPRT-TANDEM. Red and blue trajectories represent odd and even digits, respectively. (e) SAT curves of the ablation test comparing the effect of the $L_{\mathrm{multiplet}}$ and the L_{LLR}. (f) SAT curves comparing the SPRT and Neyman-Pearson test (NPT) using the same 1st-order SPRT-TANDEM network trained on the NMNIST database.

fixed-length classifier (3DResNet (Hara et al., 2017)), as baseline models. All the early-classification models share the same feature extractor as that of the SPRT-TANDEM for a fair comparison.

Hyperparameters of all the models are individually optimized with Optuna unless otherwise noted so that no models are disadvantaged by choice of hyperparameters. See Appendix G for the search spaces and fixed final parameters. After fixing hyperparameters, experiments are repeated with different random seeds to obtain statistics. In each of the training runs, we evaluate the validation set after each training epoch and then save the weight parameters if the balanced accuracy on the validation set updates the largest value. The last saved weights are used as the model of that run. The model evaluation is performed on the test dataset. During the test stage of the SPRT-TANDEM, we used various values of the SPRT thresholds to obtain a range of balanced accuracy-mean hitting time combinations for a SAT curve. If all the samples in a video are used up, the thresholds are collapsed to $a_1 = a_0 = 0$ to force a decision. To objectively compare all the models with various trial numbers, we conducted the two-way ANOVA followed by the Tukey-Kramer multi-comparison test to compute statistical significance. For the details of the statistical test, see Appendix H.

We show our experimental results below. Due to space limitations, we can only show representative results. For more details, see Appendix I. For our computing infrastructure, see Appendix J.

Nosaic MNIST (Noise + mosaic MNIST) database. We introduce a novel dataset, NMNIST, whose video is buried with noise at the first frame, and gradually denoised toward the last, 20th frame (see Appendix K for example data). The motivation to create NMNIST instead of using a preexisting time-series database is as follows: for simple video databases such as Moving MNIST (MMNIST, (Srivastava et al.)), each data sample contains too much information so that well-trained classifiers can correctly classify a video only with one or two frames (see Appendix L for the results of the SPRT-TANDEM and LSTM-m on MMNIST).

We design a parity classification task, classifying $0 - 9$ digits into an odd or even class. The training, validation, and test datasets contain 50,000, 10,000, and 10,000 videos with frames of size $28 \times 28 \times 1$ (gray scale). Each pixel value is divided by 127.5, before subtracted by 1. The feature extractor of the SPRT-TANDEM is ResNet-110 (He et al., 2016a), with the final output reduced to 128 channels. The temporal integrator is a peephole-LSTM (Gers & Schmidhuber, 2000; Hochreiter & Schmidhuber, 1997), with hidden layers of 128 units. The total numbers of trainable parameters on the feature extractor and temporal integrator are 6.9M and 0.1M, respectively. We train 0th, 1st, 2nd, 3rd, 4th, 5th, 10th, and 19th order SPRT-TANDEM networks. LSTM-s / LSTM-m and EARLIEST use peephole-LSTM and LSTM, respectively, both with hidden layers of 128 units. 3DResNet has 101 layers with 128 final output channels so that the total number of trainable parameters is in the same order (7.7M) as that of the SPRT-TANDEM.

Figure 3a and Table 1 show representative results of the experiment. Figure 3d shows example LLR trajectories calculated with the 10th order SPRT-TANDEM. The SPRT-TANDEM outperforms other baseline algorithms by large margins at all mean hitting times. The best performing model is the 10th order TANDEM, which achieves statistically significantly higher balanced accuracy than other algorithms (p-value < 0.001). The superiority of the SPRT-TANDEM over other algorithms indicates that the SPRT-TANDEM sequentially computes and minimizes the distance between the estimated LLR and the true LLR, approaching Bayes optimally if not reaches.

Table 1: Representative mean balanced accuracy (%) calculated on NMNIST. For the complete list including standard errors, see Appendix I.

Model		Mean hitting time										#trials
		2	3	4	4.37	5	6	10	15	19	19.66	
SPRT-TANDEM (proposed)	0th	92.43	97.47	98.82	99.03	99.20	99.37	99.50	99.50	99.50	99.50	100
	1st	93.81	98.04	99.07	99.21	99.34	99.46	99.50	99.50	99.50	99.50	100
	2nd	93.73	98.01	99.07	99.22	99.36	99.45	99.49	99.49	99.49	99.50	120
	10th	93.77	98.02	99.09	**99.23**	**99.37**	**99.47**	**99.51**	**99.51**	**99.51**	**99.51**	139
	19th (max)	**94.25**	**98.26**	**99.12**	**99.23**	**99.37**	99.46	99.50	99.50	99.50	99.50	100
LSTM-m		88.74	93.89	96.15		97.62	98.35	99.19	99.42	99.48		138
LSTM-s		89.01	94.13	96.47		97.91	98.43	99.28	99.45	99.52		120
EARLIEST					97.48						99.34	130
3DResNet						93.81		96.98				100

UCF101 action recognition database. To create a more challenging task, we selected two classes, handstand-pushups and handstand-walking, from the 101 classes in the UCF database. At a glimpse of one frame, the two classes are hard to distinguish. Thus, to correctly classify these classes, temporal information must be properly used. We resize each video's duration as multiples of 50 frames and sample every 50 frames with 25 frames of stride as one data. Training, validation, and test datasets contain 1026, 106, and 105 videos with frames of size 224 × 224 × 3, randomly cropped to 200 × 200 × 3 at training. The mean and variance of a frame are normalized to zero and one, respectively. The feature extractor of the SPRT-TANDEM is ResNet-50 (He et al., 2016b), with the final output reduced to 64 channels. The temporal integrator is a peephole-LSTM, with hidden layers of 64 units. The total numbers of trainable parameters in the feature extractor and temporal integrator are 26K and 33K, respectively. We train 0th, 1st, 2nd, 3rd, 5th, 10th, 19th, 24th, and 49th-order SPRT-TANDEM. LSTM-s / LSTM-m and EARLIEST use peephole-LSTM and LSTM, respectively, both with hidden layers of 64 units. 3DResNet has 50 layers with 64 final output channels so that the total number of trainable parameters (52K) is on the same order as that of the SPRT-TANDEM.

Figure 3b shows representative results of the experiment. The best performing model is the 10th order TANDEM, which achieves statistically significantly higher balanced accuracy than other models (p-value < 0.001). The superiority of the higher-order TANDEM indicates that a classifier needs to integrate longer temporal information in order to distinguish the two classes (also see Appendix D).

Table 2: Representative mean balanced accuracy (%) calculated on UCF. For the complete list including standard errors, see Appendix I.

Model		Mean hitting time										#trials
		2	2.01	2.09	3	4	5	10	15	25	49	
SPRT-TANDEM (proposed)	0th	92.92	92.94	93.00	93.38	94.06	94.66	96.04	96.83	96.91	96.91	200
	1st	93.79	93.78	93.73	93.57	93.93	94.56	95.96	96.55	96.87	96.87	200
	2nd	94.20	94.20	94.18	93.97	94.01	94.09	95.84	96.46	96.76	96.79	200
	10th	94.37	94.37	94.31	94.29	**94.77**	**95.10**	96.18	96.85	**97.12**	**97.25**	256
	49th (max)	**94.52**	**94.51**	**94.52**	**94.40**	94.36	94.51	**96.20**	**97.03**	96.96	96.72	200
LSTM-m		93.14			93.59	93.23	93.31	94.32	94.59	95.93	96.68	100
LSTM-s		90.87			92.36	92.82	93.17	93.75	94.23	95.93	96.45	101
EARLIEST			93.38	93.48								50
3DResNet									64.42	90.08		100

Spoofing in the Wild (SiW) database. To test the SPRT-TANDEM in a more practical situation, we conducted experiments on the SiW database. We use a sliding window of 50 frames-length and 25 frames-stride to sample data, which yields training, validation, and test datasets of 46,729, 4,968, and

43,878 videos of live or spoofing face. Each frame is resized to $256 \times 256 \times 3$ pixels and randomly cropped to $244 \times 244 \times 3$ at training. The mean and variance of a frame are normalized to zero and one, respectively. The feature extractor of the SPRT-TANDEM is ResNet-152, with the final output reduced to 512 channels. The temporal integrator is a peephole-LSTM, with hidden layers of 512 units. The total number of trainable parameters in the feature extractor and temporal integrator is 3.7M and 2.1M, respectively. We train 0th, 1st, 2nd, 3rd, 5th, 10th, 19th, 24th, and 49th-order SPRT-TANDEM networks. LSTM-s / LSTM-m and EARLIEST use peephole-LSTM and LSTM, respectively, both with hidden layers of 512 units. 3DResNet has 101 layers with 512 final output channels so that the total number of trainable parameters (5.3M) is in the same order as that of the SPRT-TANDEM. Optuna is not applied due to the large database and network size.

Figure 3c shows representative results of the experiment. The best performing model is the 10th order TANDEM, which achieves statistically significantly higher balanced accuracy than other models (p-value < 0.001). The superiority of the lower-order TANDEM indicates that each video frame contains a high amount of information necessary for the classification, imposing less need to collect a large number of frames (also see Appendix D).

Table 3: Representative mean balanced accuracy (%) calculated on SiW. For the complete list including standard errors, see Appendix I.

Model		Mean hitting time										#trials
		1.19	2	3	5	8.21	10	15	25	32.06	49	
SPRT-TANDEM (proposed)	0th	99.78	99.82	99.85	99.87	99.87	99.87	99.87	99.87	99.87	99.87	100
	1st	99.81	99.84	99.86	99.87	99.88	**99.89**	99.89	99.89	99.89	99.89	112
	2nd	99.82	99.86	**99.88**	**99.89**	**99.89**	**99.89**	**99.90**	**99.90**	**99.90**	**99.90**	110
	10th	**99.84**	99.87	**99.88**	99.88	99.88	99.88	99.88	99.89	99.88	99.88	107
	49th (max)	99.83	**99.88**	**99.88**	99.88	99.88	99.88	99.88	99.88	99.88	96.72	73
LSTM-m			99.77	99.80	99.80		99.81	99.83	99.85		99.88	63
LSTM-s			99.77	99.80	99.80		99.81	99.83	99.84		99.87	58
EARLIEST		99.72				99.77				99.76		30
3DResNet					98.82			98.97	98.56			5

Ablation study. To understand contributions of the $L_{\mathrm{multiplet}}$ and L_{LLR} to the SAT curve, we conduct an ablation study. The 1st-order SPRT-TANDEM is trained with $L_{\mathrm{multiplet}}$ only, L_{LLR} only, and both $L_{\mathrm{multiplet}}$ and L_{LLR}. The hyperparameters of the three models are independently optimized using Optuna (see Appendix G). The evaluated database and model are NMNIST and the 1st-order SPRT-TANDEM, respectively. Figure 3e shows the three SAT curves. The result shows that $L_{\mathrm{multiplet}}$ enables faster classification, whereas L_{LLR} leads to higher classification accuracy. The best performance is obtained by using both $L_{\mathrm{multiplet}}$ and L_{LLR}. We also confirmed this tendency with the 19th order SPRT-TANDEM, as shown in Appendix M.

SPRT vs. Neyman-Pearson test. As we discuss in Appendix A, the Neyman-Person test is the optimal likelihood ratio test with a *fixed* number of samples. The SPRT, however, reaches the accuracy with much smaller data samples in the early classification of time series, i.e., in the test with *flexible* numbers of samples. To experimentally test this prediction, we compare the SPRT-TANDEM and corresponding the Neyman-Pearson test. By using the calculated LLR trajectory, the Neyman-Pearson test classifies the entire data into two classes at each number of frames, using threshold $\lambda = 0$. Results support the theoretical prediction, as shown in Figure 3f: the Neyman-Pearson test needs a larger number of samples than the SPRT-TANDEM at the early phase with few frames, asymptotically approaches the SPRT-TANDEM at the later phase with many frames.

6 Conclusion

We presented the SPRT-TANDEM, a novel algorithm making Wald's SPRT applicable to non-i.i.d. data series without prior knowledge of the LLR. Leveraging deep neural networks and the novel loss function, LLLR, the SPRT-TANDEM minimizes the distance of the true LLR and the LLR estimated by the TANDEM formula, enabling wide application of the SPRT in real-world scenarios. Tested on the three publicly available databases, we confirm that the SPRT-TANDEM achieves statistically significantly higher accuracy over other existing algorithms with a smaller number of data. The SPRT-TANDEM enables a user to control the speed-accuracy tradeoff without additional training, opening up various potential applications where either high-accuracy or high-speed is required.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, pp. 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330701. URL https://doi.org/10.1145/3292500.3330701.

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 81–88, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273507.

A. Cabri, G. Suchacka, S. Rovetta, and F. Masulli. Online web bot detection using a sequential classification approach. In *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 1536–1540, 2018.

C. Chen, M. O. Gribble, J. Bartroff, S. M. Bay, and L. Goldstein. The Sequential Probability Ratio Test: An efficient alternative to exact binomial testing for Clean Water Act 303(d) evaluation. *J. Environ. Manage.*, 192:89–93, May 2017.

K.F. Cheng and C.K. Chu. Semiparametric density estimation under a two-sample density ratio model. *Bernoulli*, 10(4):583–604, 08 2004. doi: 10.3150/bj/1093265631.

Don Kurian Dennis, Chirag Pabbaraju, Harsha Vardhan Simhadri, and Prateek Jain. Multiple instance learning for efficient sequential data classification on resource-constrained devices. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 10976–10987, Red Hook, NY, USA, 2018. Curran Associates Inc.

R. S. Evans, K. G. Kuttler, K. J. Simpson, S. Howe, P. F. Crossno, K. V. Johnson, M. N. Schreiner, J. F. Lloyd, W. H. Tettelbach, R. K. Keddington, A. Tanner, C. Wilde, and T. P. Clemmer. Automated detection of physiologic deterioration in hospitalized patients. *J Am Med Inform Assoc*, 22(2): 350–360, Mar 2015.

R.A. Fisher. *Statistical methods for research workers*. Edinburgh Oliver & Boyd, 1925.

Felix A. Gers and Jürgen Schmidhuber. Recurrent nets that time and count. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, 3:189–194 vol.3, 2000.

Mohamed F. Ghalwash, Vladan Radosavljevic, and Zoran Obradovic. Utilizing temporal patterns for estimating uncertainty in interpretable early decision making. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 402–411, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623694. URL https://doi.org/10.1145/2623330.2623694.

I. J. Good. Studies in the History of Probability and Statistics. XXXVII A. M. Turing's statistical work in World War II. *Biometrika*, 66(2):393–396, 08 1979. ISSN 0006-3444. doi: 10.1093/biomet/66.2.393. URL https://doi.org/10.1093/biomet/66.2.393.

M. P. Griffin and J. R. Moorman. Toward the early diagnosis of neonatal sepsis and sepsis-like illness using novel heart rate analysis. *Pediatrics*, 107(1):97–104, Jan 2001.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 3154–3160, 2017.

Thomas Hartvigsen, Cansu Sen, Xiangnan Kong, and Elke Rundensteiner. Adaptive-halting policy network for early classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 101–110, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330974. URL http://doi.acm.org/10.1145/3292500.3330974.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pp. 630–645, 2016b. doi: 10.1007/978-3-319-46493-0_38. URL https://doi.org/10.1007/978-3-319-46493-0_38.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul):1391–1445, 2009.

Haidar Khan, Lara Marcuse, and Bülent Yener. Deep density ratio estimation for change point detection. *arXiv preprint arXiv:1905.09876*, 2019.

Shinichiro Kira, Tianming Yang, and Michael N. Shadlen. A neural implementation of wald's sequential probability rato test. *Neuron*, 85(4):861–873, February 2015. ISSN 08966273.

S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694.

Martin Kulldorff, Robert L. Davis, Margarette Kolczak†, Edwin Lewis, Tracy Lieu, and Richard Platt. A maximized sequential probability ratio test for drug and vaccine safety surveillance. *Sequential Analysis*, 30(1):58–78, 2011. doi: 10.1080/07474946.2011.539924. URL https://doi.org/10.1080/07474946.2011.539924.

Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *In Proceeding of IEEE Computer Vision and Pattern Recognition*, Salt Lake City, UT, June 2018.

Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216 – 231, 2019. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2019.02.023. URL http://www.sciencedirect.com/science/article/pii/S0031320319300950.

S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1942–1950, 2016.

U. Mori, A. Mendiburu, S. Dasgupta, and J. A. Lozano. Early classification of time series by simultaneously optimizing the accuracy and earliness. *IEEE Transactions on Neural Networks and Learning Systems*, 29(10):4569–4578, 2018.

Usue Mori, Alexander Mendiburu, Eamonn J. Keogh, and José Antonio Lozano. Reliable early classification of time series based on discriminating the classes over time. *Data Mining and Knowledge Discovery*, 31:233–263, 2016.

Jing Qin. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–630, 09 1998. ISSN 0006-3444.

Edward Simpson. Bayes at Bletchley Park. *Significance*, 7(2):76–80, June 2010. ISSN 17409705. doi: 10.1111/j.1740-9713.2010.00424.x. URL http://doi.wiley.com/10.1111/j.1740-9713.2010.00424.x.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. URL http://dblp.uni-trier.de/db/journals/corr/corr1212.html#abs-1212-0402.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv*.

M. Sugiyama, T. Suzuki, and Takafumi Kanamori. Density ratio estimation: A comprehensive review. *RIMS Kokyuroku*, pp. 10–31, 01 2010.

Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

Tomoyuki Suzuki, Hirokatsu Kataoka, Yoshimitsu Aoki, and Yutaka Satoh. Anticipating traffic accidents with adaptive loss and large-scale incident db. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3521–3529, 2018.

Alexander Tartakovsky, Igor Nikiforov, and Michele Basseville. *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. Chapman & Hall/CRC, 1st edition, 2014. ISBN 1439838208, 9781439838204.

Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel, and Masashi Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17: 138–155, 2009. doi: 10.2197/ipsjjip.17.138.

John W. Tukey. Comparing individual means in the analysis of variance. *Biometrics*, 5 2:99–114, 1949.

A. Wald. Sequential tests of statistical hypotheses. *Ann. Math. Statist.*, 16(2):117–186, 06 1945. doi: 10.1214/aoms/1177731118. URL https://doi.org/10.1214/aoms/1177731118.

A. Wald and J. Wolfowitz. Optimum character of the sequential probability ratio test. *Ann. Math. Statist.*, 19(3):326–339, 09 1948. doi: 10.1214/aoms/1177730197. URL https://doi.org/10.1214/aoms/1177730197.

Abraham Wald. *Sequential Analysis*. John Wiley and Sons, 1st edition, 1947. URL http://books.google.com/books?id=oVYDHHzZtdIC&printsec=frontcover&dq=editions:oVYDHHzZtdIC&hl=en&ei=P5zFTYbWNdK1twe1sfCYBA&sa=X&oi=book_result&ct=book-thumbnail&resnum=1&ved=0CCwQ6wEwAA#v=onepage&q&f=false.

Zhengzheng Xing, Jian Pei, and Philip S. Yu. Early prediction on time series: A nearest neighbor approach. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pp. 1297–1302, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

Zhengzheng Xing, Jian Pei, and Philip S. Yu. Early classification on time series. *Knowledge and Information Systems*, 31(1):105–127, April 2012. ISSN 0219-3116. doi: 10.1007/s10115-011-0400-x. URL https://doi.org/10.1007/s10115-011-0400-x.