A Quantum-Inspired Classifier for Early Web Bot Detection

Alberto Cabri, Francesco Masulli, *Senior Member, IEEE*, Stefano Rovetta, *Senior Member, IEEE*, and Grażyna Suchacka, *Senior Member, IEEE*

***Abstract*— This paper introduces a novel approach, inspired by the principles of Quantum Computing, to address web bot detection in terms of real-time classification of an incoming data stream of HTTP request headers, in order to ensure the shortest decision time with the highest accuracy. The proposed approach exploits the analogy between the intrinsic correlation of two or more particles and the dependence of each HTTP request on the preceding ones. Starting from the a-posteriori probability of each request to belong to a particular class, it is possible to assign a Qubit state representing a combination of the aforementioned probabilities for all available observations of the time series. By leveraging the underlying mathematical details of superposition and entanglement on specific subsequences, it is possible to devise a measure of membership to each class, thus enabling the system to take a reliable decision when a sufficient level of confidence is met or to continue with additional observations. The results reported in this paper objectively show the effectiveness of our quantum-inspired algorithm which outperforms other state-of-the-art approaches, including our own one based on the Sequential Probability Ratio Test.**

***Index Terms*— Quantum-inspired computing, bot detection, sequential classification, early decision, multinomial classification, multivariate sequence classification.**

I. Introduction

IN THE era of Big Data, huge volumes of varied data are collected at high velocity in several contexts, posing new challenges concerning timely recognition of anomalous or critical events.

Whenever event data are indexed on time, the relevant dataset represents a time series where each observation is somehow related to its temporal neighbors. Being able to automatically classify a sequence is a highly valuable task and even more important is the ability to label a time series with the fewest possible observations [1], [2].

Manuscript received December 11, 2021; revised April 9, 2022; accepted April 12, 2022. Date of publication April 25, 2022; date of current version May 6, 2022. This work was supported by the ICT COST Action IC1406 High-Performance Modelling and Simulation for Big Data Applications (cHiPSet). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alexey Vinel. *(Corresponding author: Alberto Cabri.)*

Alberto Cabri and Stefano Rovetta are with the Department of Informatics, Bioengineering, Robotics, and Systems Engineering (DIBRIS), University of Genoa, 16146 Genoa, Italy (e-mail: alberto.cabri@dibris.unige.it; stefano.rovetta@unige.it).

Francesco Masulli is with the Department of Informatics, Bioengineering, Robotics, and Systems Engineering (DIBRIS), University of Genoa, 16146 Genoa, Italy, and also with the Sbarro Institute for Cancer Research and Molecular Medicine, Temple University, Philadelphia, PA 19122 USA (e-mail: francesco.masulli@unige.it).

Grażyna Suchacka is with the Institute of Informatics, University of Opole, 45-040 Opole, Poland (e-mail: gsuchacka@uni.opole.pl).

Digital Object Identifier 10.1109/TIFS.2022.3170237

Many time series applications consider classification accuracy as the essential point and no particular importance is given to the speed of decision. An example of such tasks is forgeries detection on signatures, where *on-the-fly (OTF)* classification is not required whereas high accuracy is a crucial performance metric [3].

Conversely, timely decisions are an essential feature on an extrusion line in order to detect and amend possible defects before the product integrity gets compromised [4].

These simple considerations denote the dual aspects of time series analysis, that turn out in selecting different approaches to deal with the various problems. As reported in [5], the approaches can be categorized in *offline*, whenever a complete sequence should be analyzed before labeling, or *online* (also known as *on-the-fly*), if a decision must be made as soon as possible, based on incoming observations.

The latter is commonly known as *early classification* of time series [1]. Examples of such challenging problems can be found in various industrial scenarios, as shown in Table I, often related to the processing of data streams from connected devices or sensors (*Internet of Things*), which enable harvesting huge amounts of data, most frequently as a sequence of correlated observations or measures. Even video sources can be treated as a sequence of time related events, where each event is associated to a single video frame.

In all those cases, such as the ones listed in Table I, measures are collected over time and need to be analyzed in a timely manner to extract useful information about potentially critical conditions.

Time series classification models usually target the recognition rate as their main goal, but this is not sufficient for early classification or prediction where *earliness* of decision becomes a mandatory key performance indicator.

A sequence of events that, for whatever reason, may end up compromising a piece of equipment should be detected in the shortest possible time, as any delay could cause damages and unnecessary costs [6].

This paper addresses the problem of *on-the-fly* early classification for online data streams, where data are usually statistically dependent and inherently correlated over time as in the case of web bot detection, a highly critical task in cybersecurity applications, where we need to distinguish automatic web robots from human users.

Moreover we aim at labeling a temporal sequence of events using the smallest number of observations. The task is therefore an early decision problem, based on an incomplete set

TABLE I
EXAMPLE OF EARLY CLASSIFICATION PROBLEMS FOR TIME SERIES

Task	*Description*
Cyber-security	On threat detection; being able to timely discover undesired access to web sites and circumvent misuse of network resources can prevent fraudulent activities against service providers and the resulting economical and trust loss.
Disease prevention	Early recognition of a disease onset not only can save or extend patients' lives but also can guarantee a better after treatment course and limit the costs for medical care when it allows for delaying chronic pathologies.
Seizure alert	Monitoring some physiological parameters, such as oxygen saturation or tachycardia, in hospitalized patients may assist caregivers in prompt recognition of physical deterioration by raising preventative alerts.
Predictive maintenance	Identification of unusual patterns in the behavior of an industrial system can reduce both the downtime and the maintenance costs, especially when the breakdown of a component affects many dependent elements of the system. Moving from a *preventive* approach, based on service activities performed at regular intervals, to a *predictive* one, that foresees maintenance intervention only when the likelihood of breakage is above an appropriate threshold, can lead to huge savings for companies in all sectors.
Toxic leaks detection	Timely identification of toxic compounds in air is of fundamental importance for reducing the risks associated to leakage of dangerous chemicals and it is crucial to prevent operators exposure and enhance environment protection.

of events that requires *OTF* evaluation and stretches over an undefined time horizon. A critical aspect is finding the optimal trade-off between decision speed, defined in relation to the number of observations required by the trained system to take a decision, and classification accuracy, which are conflicting constraints.

To this aim, we present a new method for early classification of online data streams, inspired by the principles of quantum computing, able to classify a series of HTTP requests with outstanding accuracy and very effective in early decision making without any knowledge of sequences' time horizon. Please consider that no physical interpretation of quantum theory is implied by our algorithm despite the analogical adoption of the underlying mathematical details. The proposed approach is completely myopic and no delay cost estimate is required to force early decision because it leverages the intrinsic structure of data to propose a class label.

One important remark is that, to the best of our knowledge, no public datasets are available for bot detection, making it difficult to compare the presented results to other relevant studies; hence, the *SPRT* approach, originally discussed in [7], has been compared with the quantum-inspired algorithm to confirm its efficacy both in terms of classification metrics and decision time.

The remainder of this paper is organized as follows: Section II presents the state of the art on possible approaches to early data stream classification; Section III introduces the theoretical background on quantum computing, which is required to understand the proposed method; Section IV illustrates the validation process of the proposed method using synthetic data; Section VI describes the test problem that has been used to verify this novel approach while Section VII presents the structure of the dataset used for bot detection and the relevant features. Section VIII describes its application to the chosen classification problem, regarding the analysis of web traffic logs of a real e-commerce portal; in Section IX the experimental results are reported and commented; lastly, Section X offers concluding remarks and cues for extending the research and the possible areas of future application.

II. STATE OF THE ART

Monitoring natural and industrial processes often produce massive volumes of sequential data (data streams), usually indexed over time.

Several methods are available for modeling sequential data but Statistical models, such as *ARMA* or *ARIMA* [8], [9], aimed at time series prediction, assume the linearity of data model which means that the time series is either stationary or convertible into stationary. Most often, time series are *non-stationary* because their statistical properties vary over time and thus require data models built on training data [10], such as *Artificial Neural Networks (ANN)* [9].

Often, machine learning techniques are not suitable for sequential data because these algorithms disregard the statistical structure of a time series and are sensitive to noise, which is always present in data streams.

Many effective time series classification approaches are available in literature [2], [11], but they are not suitable for early decision: it is worth underlining that early decision is a task for analyzing data streams collected in real time and locating the earliest event that supports a reliable decision, according to a given cost function, from an incomplete set of temporally related data. It is an example of *optimal stopping theory* [12] because a given action is taken from sequential observations of a random variable, according to misclassification or delay costs.

The authors of [13] present a time series classification strategy from incomplete information, introducing the notion of *reliability* as the probability required when labeling an incomplete time series as if it were the complete data stream.

As an alternative for sequential binary classification, the authors also refer to *SPRT* [14], which is a Bayes-optimal approach, but put in evidence the *greedy* connotation of this probabilistic model, where new observations have no impact on the cumulative log-likelihood calculated from previous ones.

SPRT has also been successfully used in [7] as a probability integrator, with reject option, on the same BOT detection task proposed in this paper; it outperforms a real time binomial classification approach, presented in [15], that relies on a first-order *Discrete Time Markov Chain (DTMC)* [16], [17] to estimate the class conditional probability according to the likelihoods of initial state and the following transition patterns.

In [18], the authors address early classification for some time-sensitive applications in healthcare by means of an effective *1-Nearest Neighbor (1NN)* classifier, whose major advantage is not needing any feature selection, pre-processing, training nor configuration parameters.

In [6], early classification is made by means of probabilistic classifiers, named *Early Classification framework based on class Discriminativeness and Reliability of Predictions (ECDIRE)*, that learn the timestamps when accuracy begins to exceed class defined thresholds. The predictions are released only when timestamps match the learned values. It focuses on a set of time series of equal length, but *ECDIRE* can be utilized on variable or unknown length sequences with few minor changes.

Early odor identification by means of electronic nose sensors is addressed in [19], where the authors analyze subsequent signal chunks collected at the sensors to feed an ensemble of serially connected classifiers, with a reject option, and assign a class label when sufficient confidence is attained.

Most early classification approaches in literature, such as [2], [20], work on *univariate* time series and need the entire sequence upfront. The approaches for *multivariate* sequences become more complex because the distance measures must be able to express the correlation among features [21].

Multivariate time series cannot be treated as a collection of univariate ones, because there exists a hidden relationship among features that holds important information for the representation of real processes.

In order to leverage the correlation property in multivariate time series, [22], [23] propose *Correlation Based Dynamic Time Warping (CBDTW)*, which creates a non-overlapping segmentation of a time series by means of:

- *Principal Component Analysis (PCA)* based similarity measures to segment an unclassified sequence;
- a cost function to map each chunk to a non-negative real number and *DTW* distance to train the classifier.

Statistical analysis drives an interesting adaptive non-myopic approach [24] that requires the entire sequence be available upfront and considers a penalty factor, similarly to [19], related to decision delay and a misclassification cost to balance quality of prediction and speed of decision.

Another *early classification* model suitable for multivariate time series is presented in [25] on biomedical data, specifically in multivariate gene expression. This hybrid approach binds a generative *Hidden Markov Model (HMM)* model [26], that exploits dependencies among observations on temporal segments, and a *Support Vector Machine (SVM)* [27] for efficient discrimination of sequences.

A totally different approach to early classification of biomedical multivariate time series based on shapelets is proposed in [28]. The method, named *Multivariate Shapelet Detection (MSD)*, can achieve highly accurate classification rates analyzing up to 64% of each test sequence.

The strategy proposed in [29] looks for sub-concepts or sub-clusters that characterize the same class label. The feature variables are independently scanned to uncover the inner structure of the MTS by means of core shapelets eligible for the classifier.

In [54], the authors report various quantum algorithms that are equivalent to classical machine learning but use quantum optimization to accelerate the training process or target binary classification problems such as Quantum SVM [55] or Quantum PCA [56]. They also propose an interesting Quantum Neural Network (QNN) for time series prediction and modeling.

A true quantum algorithm for time series classification is proposed in [57] where the authors make use of quantum computing by formulating the reconstruction task as a quadratic unconstrained binary optimization (QUBO) problem, although not quantum-inspired.

To the best of our knowledge, only a very limited number of quantum-inspired classification methods are available, mainly focused on binary problems.

Binary classification is the objective of a very recent quantum-inspired method, proposed by [30], that applies quantum formalism to classical computational problems, confirming a growing interest on the topic and its promising outcomes.

A binary classifier is used to solve the *quantum state discrimination problem* introduced by Helstrom [31] considering that multiple copies of a quantum state can provide more information than the state itself. This supervised algorithm, tested on real-world and simulated binomial datasets from Penn Machine Learning Benchmark repository [32], outperforms, on average, all the most frequently used classifiers.

Another approach, described in [33], might look similar to the one in this paper: it estimates the density operators for each class and applies projective measurement on quantum states to label each data element. Though, it does not address time series, nor it exploits entanglement in classification, which confirms the innovative nature of our work.

The algorithms analyzed so far propose several possible approaches to early time series classifications, but are either too specific for particular tasks or present some limitations with regard to the number of features in the input stream or the number of classes in the target or require that the whole time series be available upfront. Our proposal gets over the aforesaid limitations by introducing a real-time classification approach that, in principle, works with any number of features and classes to determine a reliable decision at the earliest moment in time, never considering the complete sequence.

III. The Quantum Classifier

Quantum computing applies quantum-mechanical principles to data processing [34].

Those fundamental principles are:

- *Superposition* that results from linearity of the solutions of Schrödinger's equation. Adding together multiple quantum states determines another valid state and, conversely, any quantum state can be split up as sum of any number of valid states.
- *Entanglement* that occurs when the state of a composite system cannot be written as a product of states of its component systems [53]. Entangled particles can express stronger connection than their classical analogues.

The quantum bit or *qubit* is a two-state quantum system that can be in a *superposition* of state 0 and 1 at the same time, unlike the classical bits.

The quantum equivalent of classical 0 and 1 logic states is defined by the basis states of a qubit, which can be represented

in *ket* notation by the following column vectors [35]:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}.$$

The state vectors form an orthonormal basis, hence their inner products $\langle x|y\rangle$ are:

$$\langle 0|0\rangle = \langle 1|1\rangle = 1 \quad \text{and} \quad \langle 0|1\rangle = \langle 1|0\rangle = 0,$$

where the *bra* operator $\langle x|$ is the conjugate-transpose of *ket*, defined as $\langle x| = |x\rangle^{\dagger}$.

A pure qubit state $|\psi\rangle$ can be expressed as superposition of the basis states

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1}$$

where α and β, termed *probability amplitudes*, are usually complex numbers such that $|\alpha|^2$ and $|\beta|^2$ represent the probability that, after a measure, the state $|\psi\rangle$ is detected in the state $|0\rangle$ or $|1\rangle$ respectively, thus leading to

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2}$$

The *factorization* of two or more qubits [36] is called a *composite state*, computed by means of the tensor product $\otimes$, as in the following example:

$$|011\rangle = |0\rangle \otimes |1\rangle \otimes |1\rangle\,. \tag{3}$$

As sequential data streams are generally characterized by an intrinsic correlation among nearby samples, *entanglement* becomes a fundamental property to enforce the interrelationship among observations of a time series.

By definition, a state is considered entangled if it is not separable into its fundamental parts, that is, two distinct particles of a system are entangled if an item cannot be described without considering the other one. Moreover, they can be entangled even if separated by considerable distance [37].

As an example,

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00..0\rangle + |11..1\rangle)$$

represents n entangled qubits in *equal superposition*, or *CatState*; in the example, states $|00..0\rangle$ and $|11..1\rangle$ have equal probabilities $|\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$. The above equation is not separable because it is impossible to write it as a tensor product.

The term *CatState* refers to quantum superposition of two macroscopically distinct states and is derived from the hypothetical Schrödinger cat's experiment.

The behavior of a physical system can be described by a general framework defined by four postulates of quantum mechanics. Two postulates are related to superposition and measurement principles, whereas the third one describes the evolution of a closed quantum system in terms of the Schrödinger equation. Finally, the fourth one describes the admissible states for composing two or more subsystems and asserts that the state space of a composite quantum system is the tensor product (symbol $\otimes$) of the state space of its components [38].

If $|\psi_1\rangle \ldots |\psi_n\rangle$ describe the state of n isolated quantum systems, the state of the composite system is

$$|\psi\rangle = |\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle\,.$$

The last aspect to consider is how to measure the probabilities of each basis state from the resulting composite state: in a real quantum system, the measurement process alters its state, which turns into the pure state corresponding to the outcome of measurement. It can be regarded as an *interface* between the quantum and the classical domains, being the only way to extract useful information from a quantum system [38].

According to the third postulate of quantum mechanics, a collection of *measurement operators* acting linearly on the state space of the system can be used to measure a quantum state: this is commonly termed *projective measurement*.

If a system can have M possible valid outcomes, a set of $\{P_m : m \in M\}$ operators can be identified in order to obtain the probability of measuring m from the system state $|\psi\rangle$, which is

$$p(m) = \langle\psi|\, P_m^{\dagger} P_m\, |\psi\rangle\,,$$

where the symbol † indicates complex conjugation and transposition.

The operators are subject to the following condition:

$$\sum_{m\in M} P_m^{\dagger} P_m = I,$$

which ensures that all probabilities add up to 1, as per:

$$\sum_{m\in M} p(m) = \sum_{m\in M} \langle\psi|\, P_m^{\dagger} P_m\, |\psi\rangle = \langle\psi|\, I\, |\psi\rangle = 1.$$

For the two basis states $|0\rangle$ or $|1\rangle$, measurement is performed through the projectors $P_0 = |0\rangle\langle 0|$ or $P_1 = |1\rangle\langle 1|$ respectively, gathering the probabilities p_0 and p_1.

Therefore, the probability p_0 of a qubit being in state $|0\rangle$ can be obtained through *projective measurement* by the following equation

$$p_0 = \langle\psi|\, P_0\, |\psi\rangle\,. \tag{4}$$

Alternatively, whenever post-measurement state is not significant, it is possible to define a *density operator* that describes the whole system [38]

$$\rho = \sum_i P_i\, |\psi_i\rangle\, \langle\psi_i|\,, \tag{5}$$

with the following constraints:

1) Trace condition: $\mathrm{Tr}(\rho) = 1$,
2) Positivity condition: ρ is a positive operator.

The trace is a linear operator, hence in the case of a two state quantum system, the trace condition can be expanded as

$$\begin{aligned}\mathrm{Tr}(\rho) &= \mathrm{Tr}\Big(\sum_{i=0}^{1} P_i\, |\psi_i\rangle\, \langle\psi_i|\Big)\\ &= \mathrm{Tr}(P_0\, |\psi\rangle\, \langle\psi|) + \mathrm{Tr}(P_1\, |\psi\rangle\, \langle\psi|),\end{aligned}$$

which leads to the generalized probability p_i of state $|i\rangle$, expressed by

$$p_i = \mathrm{Tr}(P_i\, |\psi\rangle\, \langle\psi|). \tag{6}$$

In this paper, we propose a multinomial generalization of this setting, called **Quantum Entangled Multinomial Classifier (QEMC)**, by defining the reference orthonormal basis for N classes as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad \dots \quad |N-1\rangle = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

A pure *qubit* state $|\psi\rangle$ derives from the superposition of all basis states, according to equation

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \cdots + \alpha_{n-1} |N-1\rangle, \tag{7}$$

where $|\alpha_i|^2$ is the probability of state $|i\rangle$ and $\sum |\alpha_i|^2 = 1$.

At each time step t, let $f_i(x_t), i \in [0, N-1]$ be the class conditional probabilities of current observation x_t in the data stream.

Let

$$\alpha_{i,t} = \sqrt{f_i(x_t)}$$

then T subsequent observations of class i can be composed into a T-qubit state $|\psi_i\rangle$ by means of:

$$\begin{aligned} |\psi_i\rangle &= \alpha_i |ii\dots i\rangle \\ &= \alpha_{i,0} |i\rangle \otimes \alpha_{i,1} |i\rangle \otimes \cdots \otimes \alpha_{i,T-1} |i\rangle. \end{aligned}$$

As an example, the state $|\psi_0\rangle$ for a hypothetical class 0 at the fifth observation can be computed through:

$$\begin{aligned} |\psi_0\rangle &= \alpha_0 |00000\rangle \\ &= \alpha_{0,0} |0\rangle \otimes \alpha_{0,1} |0\rangle \otimes \alpha_{0,2} |0\rangle \otimes \alpha_{0,3} |0\rangle \otimes \alpha_{0,4} |0\rangle. \end{aligned}$$

The state $|\psi\rangle$ representing a whole data stream after T observations can be expressed as the superposition of N states, each featuring some correlation among collected observations of the relevant class, according to:

$$|\psi\rangle = |\psi_0\rangle + |\psi_1\rangle + \cdots + |\psi_{N-1}\rangle. \tag{8}$$

At every time step t, the state of quantum system $|\psi\rangle$ can be measured to provide the individual class probabilities $p_i(t), i \in [0, N-1]$ and, given a task dependent level of confidence C, make appropriate decisions as:

$$\begin{cases} i & if \quad p_i(t) \geq C \\ None & if \quad otherwise \end{cases} \tag{9}$$

If *None* is still output when the session ends, it is eventually classified as **undecided** (reject option) and considered an error.

Undecided sessions appear as a separate indicator to be considered when tuning the appropriate level of confidence C. As a matter of fact, undecided sessions represent the inability of our classifier to fulfill its purpose but, even if it is clear that the correct class cannot be designated, none of the wrong ones can be elicited as most representative without committing a mistake.

Eventually, as the probabilities $p_i(t)$ measured on state $|\psi\rangle$ are normalized, for any value of C greater than 0.5, the condition expressed by (9) becomes necessary and sufficient for a mutually exclusive decision.

TABLE II

EXAMPLE OF GENERATED PROBABILITIES THE *Session* COLUMN INDENTIFIES THE ELEMENTS IN THE SAME SERIES, WHOSE CLASS PROBABILITIES ARE REPORTED IN THE $Class_i$ COLUMNS; THE *Label* IS THE GROUND TRUTH

$Session$	$Class_0$	$Class_1$	...	$Class_{N-1}$	$Label$
0	0.49	0.01	...	0.22	0
0	0.44	0.02	...	0.19	0
1	0.03	0.12	...	0.03	1
1	0.02	0.13	...	0.02	1
1	0.02	0.21	...	0.05	1
...	...	...	...	...	...

QEMC is also characterized as a greedy algorithm, as it tries to achieve the best classification results by analyzing local probability maxima, which are not guaranteed to be optimal overall.

IV. VALIDATION ON SYNTHETIC DATA

A. Generation of Synthetic Data

The applicability of QEMC was first validated on synthetic datasets of probabilities, generated for an increasing number of classes.

The synthetic datasets simulate the results of an element-wise stream classification, therefore they contain a list of N class probabilities for a specified number of sessions having variable length up to a desired maximum number of samples.

In order to ensure a sensible bias for a specific class, every session is randomly assigned a ground truth value and, for each sample, the probability p_{true} of the **True** class is randomly taken from a continuous uniform distribution in the [0, 1) interval.

The residual probability value, $p_{res} = 1 - p_{true}$, is then used in combination with a Dirichlet distribution to generate N random values that add up to p_{res}: these likelihoods are arbitrarily allotted to each class and p_{true} is added to the **True** class.

Even if a single event line doesn't express a clear statement on which is the **True** class, the session is clearly biased and this is what the algorithm is supposed to exploit in order to make a timely decision.

Table II displays the sample structure of a N classes data stream, which is saved as a CSV file.

B. Measuring the Quantum State

In section III, the measurement process for determining the qubit state has been addressed from the theoretical viewpoint, but it is also useful to add some practical considerations about its actual implementation.

Measurement is the only way to extract useful information from a quantum system and, in the real world, it exhibits some peculiar properties that should, in principle, be replicated in software simulations. These are:

1) in a real quantum system, the measurement process alters the state of the system;
2) after measurement, the system turns into the pure state associated to the outcome of measurement.

TABLE III
NUMBER OF SESSIONS PER CLASS IN SYNTHETICALLY GENERATED DATASETS

Count for	class 0	class 1	class 2	class 3
2 classes	5053	4947	-	-
3 classes	3333	3335	3332	-
4 classes	2465	2526	2513	2496

As a consequence, in a real system, it is impossible to estimate the likelihood of all possible basis states because, once measured, the qubit no longer contains information about the other ones.

Simulation software usually measures the quantum states by generating a random number and reading the associated output, which is what quantum theory would require.

Nevertheless, in our quantum-inspired algorithm, we are not concerned about using a strictly rigorous approach to measurement and, conversely, we utilize the *density operator* defined in (6) to assess the probability, integrated over time, of each individual basis state and return the top value and its associated basis state.

Normalization of the resulting quantum state, before measurement takes place, ensures that all probabilities add up to one and therefore the classification threshold can be constrained within zero and one.

V. EXPERIMENTAL SETUP

All experiments were executed on an Intel Core i7 3.4 GHz workstation, with 16GB RAM, running Microsoft Windows 10 operating system with no CUDA support.

The software procedures were developed in Python language [39], at version 3, with additional support of the following standard distribution libraries: Numpy [40], Matplotlib [41], Scikit-Learn [42] and Pandas [43].

Extensive testing was executed on three synthetically generated datasets, containing from two to four well balanced classes respectively, totaling 10.000 sessions whose individual length does not exceed 100 observations.

The detailed breakdown of sessions by class label is reported in Table III.

A. Complexity Analysis

In its simplest implementation, the proposed algorithm would have an intractable exponential spatial complexity and cubic time complexity due to the use of tensor product. Specifically, if N is the number of classes and L_{max} is the maximum length of the time series, the spatial complexity is $O(N^{L_{max}+1})$ whereas the temporal one is $O(L_{max}^3)$. However, the finite-memory property of the addressed application problem can be exploited to bound the spatial requirements and, consequently, the time complexity. A sliding window mechanism was set up to limit the number of observations considered when calculating the entangled states. This technique was termed *peep*, for it acts as a peephole on the data stream, and it was empirically verified that *peep* values (window sizes) greater than 8, in most cases, don't bring any improvement to the overall classification scores, which tend to flatten for *peep* values greater than or equal to 4.

As an exception, in the binomial case, it is possible to compute the entangled states with a simpler procedure independent of *peep*. With more than three classes, experimental evidence shows that accuracy reaches its upper limit before exceeding the greatest bearable *peep* value, which was at the upper limit of 10 on our machine.

B. Results on Synthetic Data

The problem basically aims at optimizing two contrasting goals:

- maximize classification accuracy,
- minimize the number of observations required to make a decision.

A possible approach is based on multi-objective optimization, also known as Pareto optimization [44], to pick the optimal threshold as a function of selected indicators and optimization objectives.

Possible solutions in the *decision space* are rated according to multiple objective functions to find a setting which is optimal in some sense.

Pareto strategy defines a set of *non-dominated solutions* that cannot be improved on one objective without degrading at least one of the others.

With two objective functions, it is possible to plot the solution space and visualize the set of Pareto optimal solutions, which is also called Pareto frontier.

The performance indicators required to plot the Pareto frontier are collected by means of a grid search on the following algorithm parameters:

- the confidence level C, or decision threshold, with values $C \in \{0.55, 0.6, \ldots, 0.9, 0.95, 0.99, 0.995, 0.998\}$,
- the sliding window size with $peep \in \{4, 8\}$.

For each configuration of the grid search, the legend for parameters and summary indicators used in this paper is reported in Table IV.

Table V reports, for a peep equal to four, the parameters and their relevant metrics for those points on the Pareto front that maximize classification accuracy, minimize the number of undecided sessions or the length of the decision sequence. In order to consider the worst case, undecided sessions were included in the accuracy score.

It is evident that for low values of decision threshold, we have contrasting results depending on the aim of Pareto optimization, whereas on more selective thresholds the performance metrics are exactly the same on both sides. At low threshold values, it is possible to zero the number of unclassified sessions, with about 5% decrease in accuracy at the advantage of decision speed, even if the greatest number of sessions is classified within the second or third observation.

At higher thresholds, accuracy increase exceeds 14.5% at the cost of having 251 undecided sessions, which definitely compensates the number of erroneously classified ones of the former scenarios. Undecided sessions could be considered a limitation at first glance but, if the algorithm were analyzing a real time data feed instead of a fixed size dump file, further

TABLE IV
PARAMETERS AND SUMMARY INDICATORS LOGGED ON GRID SEARCH

Name	Description
DT	decision threshold on probabilities; valid values range from zero to one
PEEP	dimension of the sliding window
EXU	*exclude undecided* flag; when *True* the undecided sessions are removed from the calculation of accuracy
TOTSS	total number of sessions analyzed; this should be constant on all tests but it is logged to make sure all sessions have been considered
TOTUC	total number of undecided sessions
ACC	average classification accuracy, including or excluding undecided sessions according to EXU; excluding undecided sessions may take to the misleading result of 100% classification accuracy, which can be less significant because it only states that all classified sessions have been correctly recognized
F1	the F1 score, always excluding undecided sessions
PR	the precision score, always excluding undecided sessions
RE	the recall score, always excluding undecided sessions
LDS	length of the longest decision sequence; this means that at least one session required LDS steps to be classified; the average number of observations required to make a decision in the tests performed is between 3 and 5
#CL	number of classes
OBS	number of observations
ERR	classification errors
ACC-I	accuracy including undecided sessions
ACC-X	accuracy excluding undecided sessions
C70	number of decision steps required to classify the 70% of sessions
C90	number of decision steps required to classify the 90% of sessions
ADS	weighted average decision step on classified sessions

TABLE V
CLASSIFICATION RESULTS FOR 10.000 SESSIONS WITH 3 CLASSES (INCLUDING UNDECIDED SESSIONS)

Goal	DT	TOTUC	%ACC	%F1	%PR	%RE	LDS
Min LDS	0.550	0	82.84	82.84	82.84	82.84	3
Min LDS	0.998	251	97.49	100.00	100.00	100.00	27
Min TOTUC	0.800	0	87.52	87.52	87.53	87.52	5
Min TOTUC	0.998	251	97.49	100.00	100.00	100.00	27

observations might become available for undecided sessions and sooner or later make a reliable decision.

With the same settings, the classifier performance can also be assessed on an increasing number of classes, easily generated with our tool. The metrics reported in Table VI share DT $= 0.995$ and PEEP $= 4$ as common settings.

The ADS indicator is defined as the average, over the total number of sequences N of decision timestamps t_i weighted by the number of sequences classified at a given instant n_i, that is:

$$ADS = \frac{1}{N} \sum_{i=1}^{N} t_i \cdot n_i .$$

Even if the number of undecided sessions reduces the overall accuracy, its value stays steady above 97%, with very few classification errors in the binary case. If we hadn't considered unclassified streams, as if we could observe more events to support a trustful decision, we could ideally reach 100% accuracy for three and four classes and 99.98% for the binomial case respectively, with as many as 27 observations analyzed in the single worst case.

Moreover, seventy percent of classified sessions is correctly labeled within the fifth observation and *QEMC* needs only 8 steps to classify the ninety percent.

According to specific goals of the classification task, it is possible to tune the threshold to favor either accuracy or LDS, given that in all cases ADS indicator denotes high classification speed on average.

This initial experimental session pointed out an intrinsic limitation of the proposed quantum-inspired approach, allegedly due to the hardware specifications of our machine. Basically, in addition to the exponential complexity related to sequence length, also the number of classes represents a sort of barrier hampering the adoption of QEMC.

On the test machine, whose technical specifications are reported at the beginning of this section, up to 10 classes could be detected simultaneously without compromising overall system performance: alternative hierarchical approaches are possible but major changes to the proposed classification architecture are required to support two or more levels of refinement. For instance, if we were to predict possible component failures on a cyber-physical system, it would be possible to implement a first classification level capable of discriminating among the potentially affected subsystem and then pass only the involved data streams to a specialized classifier that is fine tuned for the given subsystem.

In principle, this hierarchical approach allows to cope with multinomial classification problems of any size, even on edge computers with extremely limited resources.

VI. THE BOT DETECTION PROBLEM

The application area on which we focused our experiments is cyber-security and specifically web robot detection from HTTP request server logs [5], [45], [46], similarly to the work of [47]–[49].

TABLE VI
CLASSIFICATION RESULTS FOR 10.000 SESSIONS WITH 2-3-4 CLASSES

#CL	OBS	TOTUC	ERR	%ACC-I	%ACC-X	C70	C90	LDS	ADS
2	513601	261	2	97.37	99.98	5	8	25	4.45
3	510571	271	0	97.29	100.00	5	8	27	4.64
4	513901	111	0	98.89	100.00	3	4	11	3.08

As evidenced in preceding sections, the multinomial version of our algorithm is a generalization of the binary approach, originally designed for bots classification from real-time HTTP traffic data at the web server and uses the same dataset for the experimental part in order to compare the results.

It is an early decision, multivariate, sequential classification task on a non-stationary data stream.

Web robots, or simply bots, are software programs capable of autonomously executing specific tasks over the internet, whose aim can be either good or malicious [50], [51].

These autonomous agents are pervading the net and many bots have useful purposes, such as search engine crawlers or price comparers, but some others have malicious goals, like stealing sensitive data, injecting malware or executing other harmful activities, and therefore must be identified as soon as possible to reduce their negative effects.

Usually, bots are detected through *offline* analysis of web server logs because it allows for a deeper understanding of their behavioral model thus putting in evidence the crawling differences between humans and robots [52]. Nevertheless, it would be helpful to enable web servers to tell robots and humans apart in real time and implement specific management policies that ensure the best user experience.

Concerning real time detection, to the best of our knowledge, two methods require special attention and therefore will be analyzed in detail and compared to the present quantum-inspired algorithm. The first method, described in [15], is based on transition maps and hidden Markov models, whereas the second one leverages Wald's Sequential Probability Ratio Test (SPRT) to gather information from subsequent events and eventually make a decision [7].

The solution proposed by Doran and Gokhale in [15] is an integrated method for real time and offline web robot detection that analyzes the differences between human and software visitors in the resource request patterns, considered time invariant by the authors, and imposes a minimum number of events to be observed before deciding.

Some basic concepts have to be defined for a common understanding of the remainder of this document:

- a **session**, according to common practice, is a series of requests pertaining to the same IP address and user agent string, separated by a time gap shorter than thirty minutes;
- a **request pattern** is the ordered sequence of resource requests received at the web server during a session.

Though humans and robots request different specific resources during each visit, it is not possible to characterize visitors by the mere list of requested resources. Conversely, the order by which resource files are accessed by a human visitor is inherently different from crawling algorithms, that are unlikely to exhibit human-like behaviors.

These considerations took the authors of [15] to defining a sensible taxonomy of possible resource file types, organized into 9 more general aggregations, whence they derived a semantical representation of all resource request patterns, which is capable of expressing the differences between humans and robots.

VII. THE DATASET FOR BOT DETECTION

The dataset used to test the proposed algorithm has been already utilized for [7], [46] to compare DTMC versus SPRT and contains the sequences of HTTP request headers from many different working sessions.

Each session has been manually labeled as bot (label 1) or human (label 0) generated and the classifier tries to take a reliable decision before the session ends or labels the session as *undecided*. Appropriate actions can then be taken on the *undecided* sessions according to the specific task objectives.

In order to apply the different classification models to the same bot dataset, no feature selection policy is implemented and all available features are considered, but proper pre-processing transformations are needed on the original features depending on their type.

The features, as shown in Table VII, can be divided into three categories, each requiring different pre-processing actions:

- numerical features (N) are standardized by subtracting the mean and scaling to the unit variance;
- categorical features (C) are transformed into the corresponding one-hot encoding;
- boolean features (B) are simply translated to their numerical equivalent: 0 for $False$ and 1 for $True$.

After each feature has been transformed as explained above, each HTTP request is represented as a 25-feature vector and the corresponding session becomes a series of time related vectors.

The entire dataset contains 13.395 sessions for a total number of 1.397.838 HTTP requests. The session breakdown is detailed as 6.190 sessions labeled as bots, 7.200 can be associated to human activities and 5 sessions were excluded because it was not possible to allot them to any class with sufficient confidence.

Finally, the dataset was prepared for a 10-fold cross-validation training by manually partitioning the sessions into ten roughly balanced subsets, each consisting of 619 and 720 sessions for bot and human classes respectively.

The good balancing between bot and human sessions involves that either accuracy or F1 score can be indifferently selected as representative metrics to evaluate the performance of the proposed algorithm.

TABLE VII
FEATURES LIST OF THE FEATURES AVAILABLE FOR MODEL TRAINING BEFORE PRE-PROCESSING

Feature	Type	Description
inter_arrival_time	N–int	elapsed time between two subsequent requests (in seconds)
HTTP_method	C–string	specifies the action to be performed on a given resource (e.g., GET, HEAD)
response_status	C–int	HTTP response status code (e.g., 200, 403, 404)
response_size	N–double	volume of data transfered with response (in kilobytes)
is_referrer_empty	B–bool	indicates whether the resource has been requested within another page (*True*) or not (*False*)
is_page	B–bool	indicates whether the requested resource is a page description (*True*) or an embedded object file (*False*)
is_graphic	B–bool	specifies whether the requested resource is a graphical file (*True*) or not (*False*)
is_script	B–bool	it is *True* if the requested resource is a script or a program file or *False* otherwise
is_style	B–bool	indicates whether the requested resource is a style sheet (*True*) or not (*False*)
is_datafile	B–bool	indicates whether the requested resource is an external data file (e.g., a zipped file) (*True*) or not (*False*)

VIII. THE SOFTWARE MODEL FOR BOT CLASSIFICATION

A. The Two-Stage Classification Model

The classification model can be ideally divided into two logical stages. The first stage, built upon a deep neural network, is responsible for learning the classification model and assigning an *a-posteriori* conditional class probability estimate to each individual HTTP request, independently of any other entry of the training sequences. It can eventually be replaced by any classifier which best fits the available data size to produce the aforesaid probability estimates: in the present case, the multi-layer perceptron was selected as the best option amongst the model we tested.

The second stage is based on the quantum-inspired entangled classifier described in section III designed for a two classes setting. It is noteworthy that, even if the problem is intrinsically binary, the classification outcome of the quantum module is three-state valued because a session might end before the system can take any reliable decision. Those sessions are then provisionally labeled as *undecided* and can be either neglected or included in the performance metrics computation, slightly affecting the overall results.

B. Stage 1: Probability Estimation

The neural network implements supervised learning, setting aside a fraction of the dataset for model validation and using the remaining part for training with 10-fold cross-validation. The neural network is based on the *MLPClassifier* of the *scikit-learn* toolkit [42] and it is designed as a *sigmoid* output unit on top of two 50-units hidden layers with *ReLU* activation function. This neural network configuration has heuristically proved to be the most effective among those tested for the dataset under examination. The terminal *sigmoid* layer has been selected because its output is a real number constrained between zero and one and therefore can be interpreted by the cascade stage as a probability estimate for the relevant class.

In the generalized approach for N classes, the output layer is composed by N *Softmax* units that calculate probabilities whose sum is always 1.

C. Stage 2: The Quantum Classifier Module

The second stage is the Quantum Entangled Multinomial Classifier proposed in section III for the binomial setting.

The reference orthonormal basis is defined as:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Let $\mathbf{x}_t$ be a sequence of HTTP request samples associated to a specific session and $y \in \{0, 1\}$ be the relevant ground truth, which is obviously the same across each session. The probability of request i being bot or human generated is computed by means of the Multi-Layer Perceptron at stage one and is stored in $p_{ki}, k \in \{0, 1\}$.

As explained earlier in this document, quantum entanglement can be used to express a higher level of correlation among quantum states, therefore, as each request in a session belongs to a specific class across the whole sequence and they are reasonably correlated because they are generated by the same entity, it sounds sensible to hypothesize that quantum entanglement be capable of capturing and exposing the intrinsic correlation within each session.

The probabilities of the two classes, estimated by the neural network, can be used to build a quantum entangled representation of all subsequent requests in a session. The multi-layer perceptron classifier does not capture temporal information; here we use it to assign the likelihood of each individual sample to belong to either class. As the request order in each sequence is preserved to reflect the web navigation pattern, QEMC deals with correlation by means of entanglement.

As expressed by (1), given the probabilities of the i-th observation in the sequence of length T, it can be linked to the two basis states $|0\rangle$ and $|1\rangle$, hence it is possible to compute α_i and β_i as

$$\alpha_i = \sqrt{p_{0i}} \qquad \beta_i = \sqrt{p_{1i}} \tag{10}$$

and then create the T-qubits separable states $|\psi_0\rangle$ and $|\psi_1\rangle$, according to (3), from

$$\begin{cases} |\psi_0\rangle = \alpha\,|00\ldots0\rangle = \alpha_0\,|0\rangle \otimes \alpha_1\,|0\rangle \otimes \ldots \otimes \alpha_{T-1}\,|0\rangle \\ |\psi_1\rangle = \beta\,|11\ldots1\rangle = \beta_0\,|1\rangle \otimes \beta_1\,|1\rangle \otimes \ldots \otimes \beta_{T-1}\,|1\rangle \end{cases} \tag{11}$$

The entangled state represented by a stream of n requests can be then expressed as the superposition of the two states from (11):

$$|\psi\rangle = |\psi_0\rangle + |\psi_1\rangle \tag{12}$$

In order to tell whether the current sample is due to a bot or a human, it is necessary to measure, from the entangled

state $|\psi\rangle$ by means of (4) or (6), the probabilities of the basis states $|0\rangle$ and $|1\rangle$ and compare those measurements against a properly tuned threshold C to take a decision, if enough information is contained in the given $|\psi\rangle$.

If no decision can be taken at current step, then a new $|\psi\rangle$ is computed by adding another observation until one of the measures meets the threshold or the session ends, thus leaving the output as *undecided*.

Finally, a variation of the approach described above has been tested by computing the probability amplitudes, as of (10), by means of $\alpha_i^{grade}, grade \in \mathbb{R}^+$. Even if α_i^{grade} cannot be considered a probability amplitude anymore, this option adds a degree of freedom to the proposed quantum-inspired algorithm, acting like a *fuzziness index*, that is helpful to improve the classification results and tune the output of the classifier, say for instance to reduce the number of unclassified sessions. Moreover, when $grade = 0.5$, the solution is equivalent to the formal theory.

IX. Experimental Results and Discussion

A. The Test Scenarios

The effectiveness of the proposed method can be demonstrated with respect to the most representative performance metrics for the analyzed dataset and it is helpful to compare the algorithm with one that shows optimal results on the same problem.

In this regard, the Sequential Probability Ratio Test from Wald [14] has been compared with QEMC on the same probabilities estimated by the training of stage 1. Presently, to the best of our knowledge, SPRT, proposed in [7], outperforms all other *state-of-the-art* approaches.

The main focus of the present work is not necessarily showing that the new approach outperforms the best *state-of-the-art* methods, but proving the effectiveness of a new paradigm that exploits quantum properties to take timely and reliable decisions.

The implemented two-stage model was beneficial to support the deployment of both Sequential Probability Ratio Test and the Quantum-inspired Entangled Multinomial Classifier along with the synoptical comparison of the respective results.

Three scenarios have been chosen to fairly and extensively compare the proposed and the reference approaches and possibly highlight any weaknesses in the new method. The reported results were computed as the average over more than two hundreds runs to provide a reliable assessment of our algorithm.

The number of sessions used for training has been reduced down to the 30% of entire dataset and the *peep* and *grade* hyper-parameters are relevant only to the Quantum-inspired approach. Moreover, as the SPRT algorithm has been implemented in the logarithmic form [7], the threshold reported in Table VIII are actually converted into their log. The *Validation* column represents the portion of dataset set apart for algorithm performance assessment. It is worth noting that the *peep* mechanism, though required to limit the computational effort, is a disadvantage for QEMC algorithm because it bounds the method's memory.

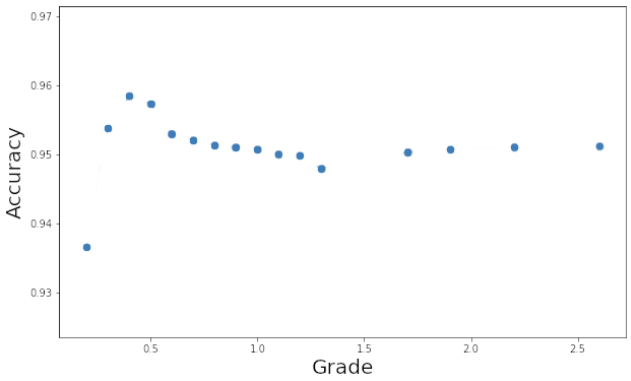


Fig. 1. Scenario A - accuracy vs grade classification accuracy at increasing values of grade.

For each scenario described in Table VIII, our aims are the minimization of the number of requests analyzed to make a decision and of the number of unclassified sessions, along with the maximization of classification accuracy, hence the same performance indicators have been considered:

- LDS: length of the decision sequence; the shorter the better,
- ACC: accuracy of classification, defined as the total number of correct assignments divided by the total number of sessions; the higher the better,
- TOTUC: total number of unclassified sessions left; to be minimized.

Pareto front plots have been generated for we need to optimize more than one objective function simultaneously at the time of decision making. These are contrasting goals because we would like to maximize accuracy whereas the length of the decision sequence and the number of unclassified sessions should be minimized. This implies that no single solution exists that can optimize all objectives but every *nondominated* solutions is Pareto-optimal and represent an acceptable solution for the problem. A solution is said *nondominated* if any improvement on an objective function implies a downgrade on the other ones. For the analysis of our results, only the solutions at extremes of the values range have been considered.

B. Scenario A

This scenario has been setup to assess the impact of different values of *grade* on the performance indicators.

The decision thresholds have been set to fixed values, identified as optimal by means of Pareto analysis, and 50% of available sessions have been set aside for model validation. The SPRT classification ends with ACC equal to 0.9422, leaving only 4 unclassified sessions and using 3 steps for LDS. As of QEMC, different values of *grade* have been tested, as shown in figure 1 but, according to the Pareto frontier plot in figure 2, the optimal points to consider for the comparison with SPRT correspond to grade 0.4, which maximizes the accuracy, and 2.6 which minimizes the length of decision sequence to the same value as SPRT.

TABLE VIII
EXPERIMENTAL SCENARIOS

Scenario	Validation	Peep	Lower Thr.	Upper Thr.	Grade
A	50%	4	0.039	0.9	$0.2 \Rightarrow 2.6$
B	70%	4	$0.05 \Rightarrow 0.25$	$0.75 \Rightarrow 0.95$	0.5
C	70%	6	0.10	0.85	$0.1 \Rightarrow 2.6$

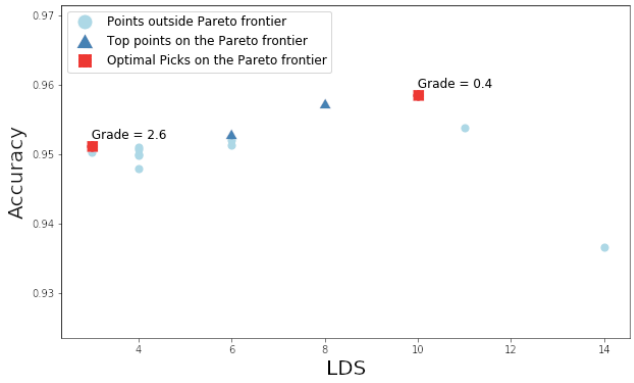


Fig. 2. Scenario A - pareto front analysis classification accuracy versus lengh of the decision sequence at variable values of Grade.

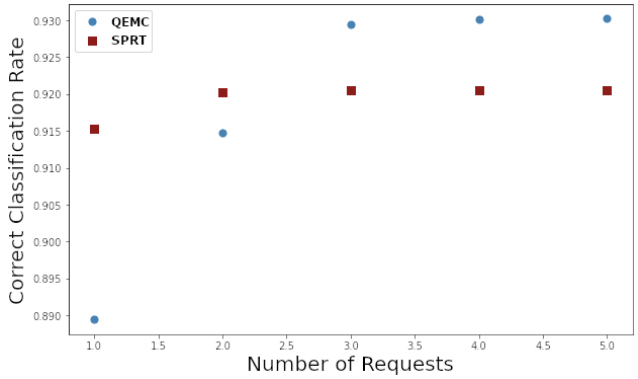


Fig. 3. Accuracy vs decision step classification accuracy achieved versus the number of requests analyzed to make a decision.

At grade 0.4, the ACC value is 0.9585, the highest for the setting, but the number of unclassified sessions is 50, which is extremely high compared to SPRT, and LDS is 10. Conversely, at grade 2.6, accuracy is only slightly less than in the previous case (ACC $=$ 0.9512, $\Delta = 0.0073$) but LDS is exactly the same as in SPRT and the number of unclassified sessions drops to zero. Nevertheless, in both cases, classification accuracy is greater than in SPRT (worst case $\Delta = 0.009$).

C. Scenario B

This scenario evaluates the classification results with regard to variable threshold values on 70% of sessions used for validation with *grade* set at 0.5, which is the default value for QEMC.

The best results for SPRT are achieved with lower and upper thresholds set to the logarithm of 0.1 and 0.85 respectively; in this configuration, ACC is 0.9205, TUC is 8 and LDS is 3. The metrics for QEMC at the best thresholds for SPRT are slightly better in accuracy (0.9302), which means that the overall number of correctly classified sessions is greater, but it might take longer to take a decision (LDS $=$ 5), even if in both cases the 90% of sessions is classified at the first step, and the number of unclassified sessions is almost double (TUC $=$ 15).

For the current setting, figure 3 visualizes the rate of correctly classified sessions for the two methods: SPRT identifies a greater percentage at the first two requests but no great improvement is achieved on the third and last step. Conversely, QEMC takes over at the third request and the overall performance is nearly 1% better than SPRT.

The best threshold pair for QEMC is 0.25 for the lower and 0.95 for the upper threshold where, despite even greater values of LDS (7) and TUC (23), the accuracy sensibly rises to 0.9527 ($\Delta = 0.0322$ versus best SPRT) and the 90% of sessions is classified within the second step. For these threshold values, the accuracy of SPRT is slightly lower (0.9204) than the best case, but the number of unclassified sessions decreases to 4 while maintaining the same LDS value.

D. Scenario C

The third scenario compares the performance indicators when varying *grade* in the threshold setting that is best for SPRT and with *peep* $=$ 6, which should improve the accuracy of QEMC by considering more samples in the decision process. Even in this case the results for SPRT are ACC is 0.9205, TUC is 8 and LDS is 3 because the *peep* mechanism only applies to QEMC, which conversely improves its classification performance depending on the Pareto optimal values of *grade*.

The optimal value to maximize accuracy is 0.2, as shown figure 5, where accuracy is 0.9589, a bit higher ($\Delta = 0.0004$) than in Scenario A with *peep* at 4, showing that it is possible to achieve better classification rates by considering more samples. This is paid for in terms of LDS, that grows to 15, TUC that spikes to 91 and on the number of steps required to classify the 90% of the sessions which becomes 3.

On the other side, the optimal value of *grade* to minimize LDS is 2.4, which not only requires at most 2 samples to take a reliable decision but also allows to achieve zero on the total number of unclassified sessions. The good point here is that accuracy is only 10^{-4} worse than for SPRT, with only 1 request needed to classify 90% of the sessions in both cases.

The three scenarios proposed above are representative of the various combinations of post-training hyper-parameters and expose both the pros and cons of the novel quantum-inspired approach.

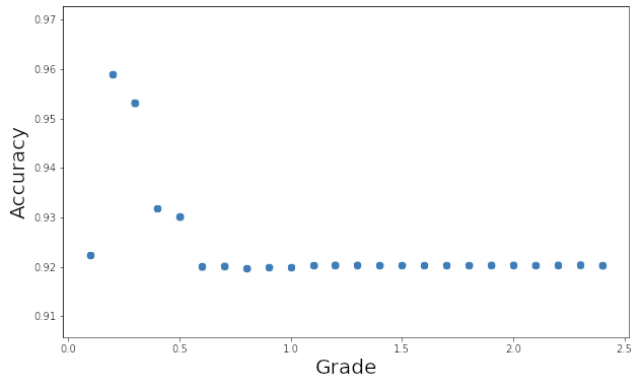


Fig. 4. Scenario C - accuracy vs grade classification accuracy at increasing values of grade.

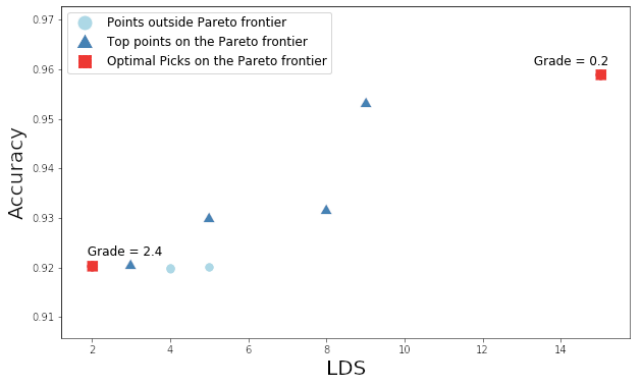


Fig. 5. Scenario C - pareto front analysis classification accuracy versus lengh of the decision sequence at variable values of grade.

Classification accuracy for QEMC can be sensibly boosted by properly selecting the *peep* and *grade* values, at the same threshold conditions, by means of Pareto analysis. Moreover the same parameters can be used to achieve particular objectives, such as zero unclassified sessions or a shorter decision sequence, while preserving the performance indicators that, in the worst case, are fairly equal. In fact, by tuning *peep* and *grade*, it is possible to increase the convergence speed of the classification algorithm and reduce the number of requests needed to take a decision to even less than SPRT.

It is worth noting that a reduction in the training size of the dataset has a smaller impact on classification accuracy for QEMC than for SPRT, in the same setting: experimental evidence shows that, with a validation ratio of 50%, accuracy is 0.9573 for QEMC and 0.9421 for SPRT whereas, with 30% of the sessions used for training, the relevant values are 0.9535 and 0.9204 respectively. Hence $\Delta_{QEMC} = 0.0038$ and $\Delta_{SPRT} = 0.0217$, which is nearly 6 times greater than the former.

Another important consideration is related to the *peep* value: the adoption of such mechanism is imposed by the computational performance downgrade on long sequences when the decision process requires to consider many requests to meet the desired confidence level. However, regardless of the length of a session, the number of samples that have to be taken into account, as shown in sections IV and IX, it is often limited to 4 to 6 samples. Greater *peep* values do not bring any benefit to the classification performance but increase the computational effort.

Finally, while SPRT is designed as a binary classifier and requires a modified approach to be applied in a multi-class problem, the QEMC method is natively suited for multinomial problems by simply expanding the orthonormal basis through the addition of further basis states.

X. Conclusion

In the present paper, we analyzed the general structure of a temporal sequence of data and pointed out the benefits of real time classification of non stationary data streams, underlining its application in cyber-security with *on-the-fly* bot detection.

We introduced QEMC, a novel quantum-inspired multinomial classifier for early detection of significant events on time series, that has been validated in a synthetic experimental setting to confirm the motivating results obtained with its binary version applied to bot detection.

The proposed technique relies on superposition and entanglement to integrate the class probability of each individual event in the time series, estimated by an upstream stage, and produce an overall score, with reject option, capable of supporting trustful decisions even in case of a limited number of events.

Our method has been successfully compared with another effective bot detection approach, namely SPRT, and its results have been analyzed with reference to the contrasting objectives of classification accuracy, number of undecided sessions and speed of decision.

The extensive experimental studies, tested on traffic streams from an actual Polish e-commerce server, showed that SPRT is able to detect, in real time, over 90% of all bots and is especially powerful given a very limited number of observations, despite it requires no minimum quantity of HTTP requests to be observed before making a decision.

Nonetheless, our innovative quantum-inspired multinomial classifier for early detection of significant events on time series can produce better overall scores and is similarly capable of supporting trustful decisions even in case of a limited number of events, both in the binary and in the multinomial setting.

The results were analyzed with reference to the contrasting objectives of classification accuracy, number of undecided sessions and speed of decision, showing that the proposed quantum-inspired algorithm, in our opinion, natively covers an area of application (non-stationary data stream classification) that so far has not yet found reliable and performing approaches.

This paper demonstrates the effectiveness of the proposed algorithm that, compared to other approaches, was proven to outperform not only SPRT but also, by transitive property, other very powerful *state-of-the-art* techniques.

Moreover, the proposed approach represents a complete real time classification framework for a critical application, such as bot detection, and can easily be integrated, as a *plug-in*, in a web architecture.

With regard to the methods analyzed in section II, some additional notes are worth reporting to highlight the advantages and disadvantages of current implementation of the new approach:

1) QEMC is tolerant against non-standardized numerical features, which is usually considered a compelling transformation for machine learning tasks;
2) with QEMC, it is possible to dramatically reduce the number of training sequences with no significant decrease of classification scores;
3) in current configuration of the classification framework, solutions are not interpretable, therefore some areas of application might be precluded to QEMC;
4) no estimate on reliability of decisions is currently available in QEMC;
5) dependencies on *grade* parameter have not yet been explored in depth, but could open the way to a *fuzzy* flavor of the classifier.

In our opinion, considering the interesting results achieved with this initial formulation of QEMC, the last three items represent interesting areas of investigation, where near future research should be directed. We also believe that the proposed algorithm might open the way to new approaches for time series prediction and clustering, but so far we do not envisage any sensible evolution.

Replacement of the ANN with explainable ways to compute the probability estimates of observations might also open new perspectives for the quantum-inspired technique, especially if accompanied by a measure of decision reliability.

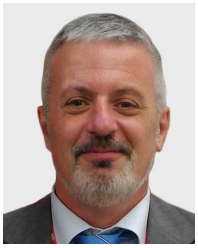
ACKNOWLEDGMENT

The authors would like to thank Paolo Solinas for his precious support in reviewing some technical aspects of quantum theory.

REFERENCES

[1] T. Santos and R. Kern, "A literature survey of early time series classification and deep learning," in *Proc. SAMI iKNOW*, 2016, pp. 1–7.
[2] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *ACM SIGKDD Explorations Newslett.*, vol. 12, no. 1, pp. 40–48, Jun. 2010.
[3] A. Hassaïne and S. Al-Maadeed, "An online signature verification system for forgery and disguise detection," in *Proc. Neural Inf. Process.*, vol. 7666. Berlin, Germany: Springer, 2012, pp. 552–559.
[4] A. Oleff, B. Küster, M. Stonis, and L. Overmeyer, "Process monitoring for material extrusion additive manufacturing: A state-of-the-art review," *Prog. Additive Manuf.*, vol. 6, no. 4, pp. 705–730, May 2021.
[5] S. Rovetta, A. Cabri, F. Masulli, and G. Suchacka, "Bot or not? A case study on bot recognition from web session logs," in *Quantifying Processing Biomedical and Behavioral Signals*, vol. 103. Cham, Switzerland: Springer, 2019, pp. 197–206.
[6] U. Mori, A. Mendiburu, E. Keogh, and J. A. Lozano, "Reliable early classification of time series based on discriminating the classes over time," *Data Mining Knowl. Discovery*, vol. 31, no. 1, pp. 233–263, Jan. 2017.
[7] G. Suchacka, A. Cabri, S. Rovetta, and F. Masulli, "Efficient on-the-fly web bot detection," *Knowl.-Based Syst.*, vol. 223, Jul. 2021, Art. no. 107074.
[8] P. J. Brockwell and R. A. Davis, Eds., *Introduction to Time Series and Forecasting* (Springer Texts in Statistics). New York, NY, USA: Springer, 2002.
[9] R. Adhikari and R. K. Agrawal, "An introductory study on time series modeling and forecasting," Feb. 2013, *arXiv:1302.6613*.
[10] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003.
[11] S. Laxman and P. S. Sastry, "A survey of temporal data mining," *Sadhana*, vol. 31, no. 2, pp. 173–198, Apr. 2006.
[12] G. Peskir and A. N. Shiriaev, *Optimal Stopping and Free-Boundary Problems* (Lectures in Mathematics ETH Zürich). Boston, MA, USA: Birkhäuser-Verlag, 2006.
[13] N. Parrish, H. S. Anderson, M. R. Gupta, and D. Y. Hsiao, "Classifying with confidence from incomplete information," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 3561–3589, Dec. 2013.
[14] A. Wald, "Sequential tests of statistical hypotheses," *Ann. Math. Statist.*, vol. 16, no. 2, pp. 117–186, Jun. 1945.
[15] D. Doran and S. S. Gokhale, "An integrated method for real time and offline web robot detection," *Expert Syst.*, vol. 33, no. 6, pp. 592–606, Dec. 2016.
[16] F. Biagini and M. Campanino, "Discrete time Markov chains," in *Elements Probability and Statistics*, vol. 98. Cham, Switzerland: Springer, 2016, pp. 81–87.
[17] N. Privault, *Understanding Markov Chains: Examples and Applications* (Springer Undergraduate Mathematics Series), 2nd ed. Singapore: Springer, 2018.
[18] Z. Xing, J. Pei, and P. S. Yu, "Early classification on time series," *Knowl. Inf. Syst.*, vol. 31, no. 1, pp. 105–127, Apr. 2012.
[19] N. Hatami and C. Chira, "Classifiers with a reject option for early time-series classification," Dec. 2013, *arXiv:1312.3989*.
[20] Z. Xing, J. Pei, and P. S. Yu, "Early prediction on time series: A nearest neighbor approach," in *Proc. 21st Int. Conf. Artif. Intell. (IJCAI)*, 2009, pp. 1297–1302.
[21] H. Anderson, N. Parrish, and M. R. Gupta, "Early time-series classification with reliability guarantee," Sandia National Lab, Albuquerque, NM, USA, Tech. Rep. SAND2012-7379C 480398, 2012.
[22] Z. Bankó and J. Abonyi, "Correlation based dynamic time warping," in *Proc. 8th Int. Symp. Hung. Researchers Comput. Intell. Inf.*, 2007.
[23] Z. Bankó and J. Abonyi, "Correlation based dynamic time warping of multivariate time series," *Expert Syst. Appl.*, vol. 39, no. 17, pp. 12814–12823, Dec. 2012.
[24] A. Dachraoui, A. Bondu, and A. Cornuéjols, "Early classification of time series as a non myopic sequential decision making problem," in *Machine Learning and Knowledge Discovery in Databases*, vol. 9284. Cham, Switzerland: Springer, 2015, pp. 433–447.
[25] M. F. Ghalwash, D. Ramljak, and Z. Obradović, "Early classification of multivariate time series using a hybrid HMM/SVM model," in *Proc. IEEE Int. Conf. Bioinform. Biomed.*, Philadelphia, PA, USA, Oct. 2012, pp. 1–6.
[26] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[27] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer Series in Statistics), 2nd ed. New York, NY, USA: Springer, 2009.
[28] M. F. Ghalwash and Z. Obradovic, "Early classification of multivariate temporal observations by extraction of interpretable shapelets," *BMC Bioinf.*, vol. 13, no. 1, p. 195, Dec. 2012.
[29] G. He, Y. Duan, R. Peng, X. Jing, T. Qian, and L. Wang, "Early classification on multivariate time series," *Neurocomputing*, vol. 149, pp. 777–787, Feb. 2015.
[30] G. Sergioli, R. Giuntini, and H. Freytes, "A new quantum approach to binary classification," *PLoS ONE*, vol. 14, no. 5, May 2019, Art. no. e0216224.
[31] C. W. Helstrom, "Quantum detection and estimation theory," *J. Statist. Phys.*, vol. 1, no. 2, pp. 231–252, 1969.
[32] R. S. Olson, W. L. Cava, P. Orzechowski, R. J. Urbanowicz, and J. H. Moore, "PMLB: A large benchmark suite for machine learning evaluation and comparison," Mar. 2017, *arXiv:1703.00512*.
[33] P. Tiwari and M. Melucci, "Towards a quantum-inspired binary classifier," *IEEE Access*, vol. 7, pp. 42354–42372, 2019.
[34] E. Rieffel and W. Polak, *Quantum Computing: A Gentle Introduction* (Scientific and Engineering Computation). Cambridge, MA, USA: MIT Press, 2011.
[35] V. Moret-Bonillo, "Can artificial intelligence benefit from quantum computing?" *Prog. Artif. Intell.*, vol. 3, no. 2, pp. 89–105, Mar. 2015.
[36] A. Ekert, P. M. Hayden, and H. Inamori, "Basic concepts in quantum computation," in *Coherent Atomic Matter Waves*, vol. 72, R. Kaiser, C. Westbrook, and F. David, Eds. Berlin, Germany: Springer, 2001, pp. 661–701.
[37] E. G. Rieffel and W. Polak, "An introduction to quantum computing for non-physicists," Jan. 1998, *arXiv:quant-ph/9809016*.

[38] E. B. Guedes, F. M. de Assis, and R. A. C. Medeiros, "Fundamentals of quantum information processing," in *Quantum Zero-Error Information Theory*. Cham, Switzerland: Springer, 2016, pp. 7–26.
[39] G. Van Rossum and F. L. Drake, Jr., *Python Reference Manual*. Amsterdam, The Netherlands: Centrum voor Wiskunde en Informatica Amsterdam, 1995.
[40] C. R. Harris *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.
[41] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, May 2007.
[42] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2012.
[43] W. McKinney, "Data structures for statistical computing in Python," in *Proc. 9th Python Sci. Conf.*, Austin, TX, USA, vol. 445, 2010, pp. 51–56.
[44] V. Pareto, *Manuel d'économie Politique*. Geneva, Switzerland: Librairie Droz, 1981.
[45] G. Suchacka, "Analysis of aggregated bot and human traffic on e-commerce site," in *Proc. Conf. Comput. Sci. Inf. Syst.*, Sep. 2014, pp. 1123–1130.
[46] A. Cabri, G. Suchacka, S. Rovetta, and F. Masulli, "Online web bot detection using a sequential classification approach," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun., IEEE 16th Int. Conf. Smart City, IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 1536–1540.
[47] A. Lagopoulos, G. Tsoumakas, and G. Papadopoulos, "Web robot detection in academic publishing," Nov. 2017, *arXiv:1711.05098*.
[48] A. Stassopoulou and M. D. Dikaiakos, "Web robot detection: A probabilistic reasoning approach," *Comput. Netw.*, vol. 53, no. 3, pp. 265–278, Feb. 2009.
[49] P.-N. Tan and V. Kumar, "Discovery of web robot sessions based on their navigational patterns," *Data Mining Knowl. Discovery*, vol. 6, no. 1, pp. 9–35, 2002.
[50] I. Zeifman. (Jan. 2017). *Bot Traffic Report 2016*. [Online]. Available: https://www.incapsula.com/blog/bot-traffic-report-2016.html
[51] G. Buehrer, J. Stokes, K. Chellapilla, and J. Platt, "Classification of automated web traffic," in *Weaving Services and People on the World Wide Web*, Berlin, Germany: Springer-Verlag, Jan. 2009.
[52] G. Suchacka and M. Sobkow, "Detection of internet robots using a Bayesian approach," in *Proc. IEEE 2nd Int. Conf. Cybern. (CYBCONF)*, Jun. 2015, pp. 365–370.
[53] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2010, p. 96.
[54] D. Emmanoulopoulos and S. Dimoska, "Quantum machine learning in finance: Time series forecasting," Feb. 2022, *arXiv:2202.00599*.
[55] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Phys. Rev. Lett.*, vol. 113, no. 13, Sep. 2014, Art. no. 130503.
[56] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Phys.*, vol. 10, no. 9, pp. 631–633, Jul. 2014.
[57] S. Yarkoni, A. Kleshchonok, Y. Dzerin, F. Neukart, and M. Hilbert, *Semi-Supervised Time Series Classification Method for Quantum Computing* (Quantum Machine Intelligence), New York, NY, USA: Springer, Apr. 2021.



Alberto Cabri received the degree in electronic engineering from the University of Genoa, Italy, in 1992, and the Ph.D. degree in computer science and systems engineering. He is currently a qualified Teacher of computer science with the Public Secondary Schools, Genoa, Italy. He is also a Professional Engineer with the University of Genoa. His research focuses on machine learning and he has developed an innovative quantum inspired algorithm for multivariate time series classification.

Francesco Masulli (Senior Member, IEEE) is currently a Full Professor of computer science with the University of Genoa, Italy, and an Adjunct Professor with Temple University, Philadelphia, PA, USA. He held visiting positions at Radboud University, Njmegen, The Netherlands; the International Computer Science Institute, Berkeley, CA, USA; and the I3S Laboratory, University of Nice Sophia Antipolis, France. He is the author of more than 250 papers in machine learning, neural networks, clustering, fuzzy systems, and their applications. He serves as the Chair for IEEE Italy Section Computational Intelligence Society Chapter.

Stefano Rovetta (Senior Member, IEEE) is currently an Associate Professor of computer science with the University of Genova, Italy. He has authored more than 170 scientific articles in machine learning, neural networks, clustering, fuzzy systems, and bioinformatics. He is a member of the Italian Neural Network Society, the European Neural Network Society, and the European Society for Fuzzy Logic and Technology. He received the 2008 Pattern Recognition Society Award. He was the chair of international conferences.

Grażyna Suchacka (Senior Member, IEEE) received the M.Sc. degree in computer science, the M.Sc. degree in management, and the Ph.D. degree (Hons.) in computer science from the Wrocław University of Science and Technology, Poland. She is currently an Assistant Professor with the Institute of Informatics, University of Opole, Poland. Her research interests include data analysis and modeling, data mining, machine learning, and quality of web service with special regard to bot detection and electronic commerce support.