Efficient and Privacy-Preserving Decision Tree Classification for Health Monitoring Systems

Jinwen Liang, *Graduate Student Member, IEEE*, Zheng Qin, *Member, IEEE*, Liang Xue, Xiaodong Lin, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Due to the increasing healthcare costs and the advance of wireless technology, health monitoring systems have been widely adopted recently. In health monitoring systems, a hospital outsources a clinical decision model to a cloud service provider, which receives biomedical data from remote clients and produces clinical decisions based on the outsourced model. Due to critical privacy concerns, both the clinical decision model and biomedical data should be protected. In this article, we propose an efficient and privacy-preserving decision tree (PPDT) classification scheme for health monitoring systems. Specifically, we first transform a decision tree classifier (i.e., the clinical decision model) into the Boolean vectors. Then, we leverage symmetric key encryption to encrypt the Boolean vectors as encrypted indices. The PPDT classification is achieved by searching the encrypted indices with encrypted tokens. We formulate a leakage function and provide the security definition and simulation-based proof for PPDT. The performance analyses demonstrate that PPDT is very efficient in terms of computation, communication, and storage. Experimental evaluations show that PPDT only requires microsecond-level execution time, kilobyte-level communication costs, and kilobyte-level storage costs on the test data set.**

***Index Terms*—Cloud computing, decision tree classification, health monitoring systems, symmetric key encryption.**

Manuscript received April 14, 2020; revised December 29, 2020; accepted March 9, 2021. Date of publication March 17, 2021; date of current version August 6, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant U20A20174, Grant 61772191, Grant 61902123, and Grant 62002112; in part by the China Scholarship Council under Grant 201806130132; in part by the National Key Research and Development Projects under Grant 2018YFB0704000; in part by the Science and Technology Key Projects of Hunan Province under Grant 2015TP1004, Grant 2018TP2023, and Grant 2019WK2072; in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada; in part by the China Postdoctoral Science Foundation under Grant 2020M672488; in part by the Natural Science Foundation of Hunan Province under Grant 2020JJ5085; and in part by the Science and Technology Key Projects of Changsha City under Grant kq2004025, Grant kq2004027, and Grant kq2006029. *(Corresponding author: Zheng Qin.)*

Jinwen Liang is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China, and also with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: jimmieleung@hnu.edu.cn).

Zheng Qin is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: zqin@hnu.edu.cn).

Liang Xue and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: liang.xue@uwaterloo.ca; sshen@uwaterloo.ca).

Xiaodong Lin is with the School of Computer Science, University of Guelph, Guelph, ON N1G 2W1, Canada (e-mail: xlin08@uoguelph.ca).

Digital Object Identifier 10.1109/JIOT.2021.3066307

I. Introduction

With the growing cases of chronic diseases, more and more patients have to test their health conditions constantly at hospitals, which leads to skyrocketing costs for healthcare systems [1]. To reduce the healthcare costs and improve the healthcare quality, health monitoring systems, which are often built by utilizing decision tree classification, help patients to test their health conditions periodically [2]. Coupled with recent advances of wearable devices and mobile communication networks [3], health monitoring systems work as follows: a hospital first utilizes the decision tree classification technique to produce a clinical decision model, and later tests clients' biomedical data collected from wearable devices and provides decisions for clients based on the model [4], [5]. To further reduce the costs on the hospital side and enable practical deployment, the hospital often outsources the health monitoring services to a cloud server, which brings prominent benefits for both clients and the hospital, such as ubiquitous access, ease of management, and scalability [6].

Despite the well-known benefits, outsourcing health monitoring services to a semitrusted cloud also arises critical privacy concerns [7]–[9]. On the hospital side, since the hospital may invest a large number of resources to gather a sensitive biomedical data set and train the clinical decision model, the model is valuable intellectual property, which brings commercial benefits to the hospital. Thus, there is a demand for the hospital to protect the content of clinical decision model when outsourcing health monitoring services to the cloud service provider. On the clients' side, both the physiological features and the clinical decision are sensitive biomedical data, because the accidental leakage of either information may reflect the clients' health condition and lead to serious issues. For instance, if a client has a certain chronic disease, the exposure of health condition deterioration may increase the health insurance costs to the client. With the aforementioned privacy concerns, both the clinical decision model and the biomedical data should be concealed from the cloud service provider in health monitoring systems.

To protect the confidentiality of both clinical decision models and biomedical data, several privacy-preserving decision tree (PPDT) classification schemes have been proposed [10]–[20]. Most of the existing schemes are constructed based on homomorphic encryption (HE) [10]–[14] and secure multiparty computation (MPC) [15]–[19]. HE-based schemes enable PPDT classification by

homomorphically encrypting the clinical decision model and data, which may incur prohibitive computational overheads [10]–[13]. MPC-based schemes enable multiple parties jointly and privately classify data according to decision trees, but they may lead to expensive communication costs [15]–[19]. To reduce the computation and communication overheads, Liang *et al.* proposed a secure decision tree classification scheme by utilizing symmetric key encryption [20]. Although the scheme in [20] achieves $\mathcal{O}(1)$ computational complexity, it constructs huge indices, whose size is exponential to the size of the decision tree, for PPDT classification, which incurs heavy storage overheads. In summary, two main challenges should be addressed when designing PPDT classification schemes for health monitoring systems.

1) *Confidentiality:* Both biomedical data and clinical decision model should be protected against the cloud service provider.
2) *Efficiency:* The computation, communication, and storage costs should be low.

In this article, we address the aforementioned two challenges simultaneously and propose an efficient and PPDT classification scheme for health monitoring systems. First, we utilize the scheme in [21] to extract rules from decision trees by traversing all decision paths from the root node to the leaf nodes. Then, we build indices for these rules. The indices are constructed from the Boolean vectors, whose size is polynomial to the number of internal nodes, leaf nodes, and input domains. With such indices, the decision tree classification process achieves $\mathcal{O}(1)$ computational complexity, as well as high communication and storage efficiency. After that, we propose PPDT, which incorporates symmetric key encryption, pseudorandom function, and pseudorandom permutation, to enable PPDT classification by encrypting the aforementioned indices. Accordingly, PPDT not only protects the confidentiality of both the clinical decision model and biomedical data but also achieves computation, communication, and storage efficiency for health monitoring systems. The contributions of this article are summarized as follows.

1) We propose an efficient and PPDT classification scheme for health monitoring systems. First, we transform decision tree classifiers into the Boolean vectors, which are indices that enable $\mathcal{O}(1)$ computational complexity for decision tree classification. With such Boolean vectors, PPDT significantly improves computation, communication, and storage efficiency simultaneously. By utilizing symmetric key encryption, pseudorandom functions (Prf), and pseudorandom permutations (Prps) to protect the confidentiality of clinical decision models and biomedical data, PPDT significantly reduces the computational costs due to the adoption of low-complexity cryptographic primitives.
2) We formulate a security definition and give a simulation-based security proof for PPDT. First, we identify a leakage function $\mathcal{L}$, which includes the size pattern, search pattern, and access pattern of PPDT. Then, we formulate the $\mathcal{L}$-security definition, which is defined based on the leakage function $\mathcal{L}$. Finally, we provide a simulation-based security proof to demonstrate that PPDT captures the $\mathcal{L}$-security definition. Namely, both the clinical decision model and biomedical data are well protected.
3) We conduct performance analyses and evaluations for PPDT. We analyze the computational costs and index sizes of PPDT and the scheme in [20] (SDTC). Despite both PPDT and SDTC are with $\mathcal{O}(1)$ computational complexity, the comparison results show that PPDT requires lower computational costs and smaller index sizes than SDTC. The experimental evaluations in the Breast-Cancer-Wisconsin data set also illustrate the performance advantages of PPDT. The performance evaluations demonstrate that: a) the computational complexity of PPDT is $\mathcal{O}(1)$; b) PPDT only requires microseconds-level execution time, kilobyte-level communication costs, and kilobyte-level storage costs for achieving PPDT classification; and c) the performance (including computation, communication, and storage efficiency) of PPDT is orders of magnitudes boosted than SDTC.

The remainder of this article is organized as follows. Section II describes the related work. Section III provides the system model, threat model, and design goals. Section IV illustrates the preliminaries. Section V describes the construction of PPDT. Section VI formulates the leakage function and security definition, and provides a simulation-based security proof. Section VII analyzes and evaluates the performance of PPDT. Section VIII concludes this article.

II. Related Work

Driven by prominent advantages, such as high accuracy, ease of deployment, and efficient evaluation, data classification techniques have been used in many application fields, such as transportation [22]–[26], malware detection [27], and healthcare [28]–[30]. In health monitoring systems, decision tree classification is often utilized to build a clinical decision model, which is further used to make decisions to biomedical features collected by wearable devices [31]. To provide services to remote clients and reduce the costs at the hospital side, health monitoring systems often require a hospital to submit the clinical decision model to a cloud services provider, and later provide health monitoring service to remote clients [19], [20]. Since a cloud service provider is not fully trusted, both the hospital and clients may worry about the privacy leakage of both the clinical decision model and biomedical data [32].

To protect the confidentiality of the clinical decision model and biomedical data in health monitoring systems, a significant amount of PPDT classification schemes have been proposed based on cryptographic tools, such as fully HE (FHE) [10], [11], additive HE (AHE) [12], [13], garbled circuit (GC) [17], [18], oblivious transfer (OT) [12], [16]–[18], secret sharing (SS) [16], [19], searchable symmetric encryption (SSE) [20], etc. We roughly divide existing PPDT classification schemes into three categories, i.e., HE-based schemes, MPC-based schemes, and SSE-based schemes.

TABLE I
DIFFERENCES BETWEEN EXISTING SCHEMES

Schemes	[11]	[12]	[13]	[15]	[16]	[17]	[18]	[19]	[20]	PPDT
Security Paradigms	FHE	AHE + OT	AHE	GC	OT + SS	GC + OT	OT + GC + AHE	SS	SSE	SSE
Low Comp. Costs	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
Low Comm. Costs	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓
Low Storage Costs	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	✗	✓

HE-Based Schemes: Most of the HE-based schemes consider a hospital-client setting for health monitoring systems [10]–[13]. In this setting, the hospital's clinical decision tree model is required to be protected from the client, while the client's biomedical features and clinical predictions are required to be protected from the hospital. HE-based schemes protect the privacy of the clinical model and biomedical data by utilizing FHE [10], [11] or AHE [12], [13]. Although HE-based schemes enable PPDT classification for health monitoring systems, these schemes may face high computational costs due to high-complexity homomorphic operations [10]–[13]. To achieve efficient health monitoring services, it is desirable to design a scheme with low computational costs. Namely, achieving sublinear computational complexity and avoiding cryptographic tools with expensive computational costs.

MPC-Based Schemes: Considering a system model with multiple noncolluding hospitals or multiple noncolluding clients, most of the MPC-based schemes evaluate the prediction interactively and collaboratively. MPC-based schemes protect the privacy of both clinical decision model and biomedical data by utilizing tailored GC [15], SS [16], [19], or hybrid cryptographic tools that combine OT, SS, GC, and AHE [16]–[18]. Although MPC-based schemes avoid using high-complexity encryption, these schemes are designed based on a multiple party noncollusion security assumption and incur prohibitive communication costs due to collaboratively evaluation. To achieve real-time health monitoring services, it is important to design a novel scheme with low communication costs, i.e., avoiding MPC techniques.

SSE-Based Schemes: To improve the computational and communication efficiency for health monitoring systems, Liang *et al.* consider a system model that involves a hospital, a client, and a cloud service provider. In this setting, the hospital outsources the clinical decision model to a cloud service provider and, thus, the clinical decision model and biomedical data should be protected against the cloud service provider [20]. Liang *et al.* extracted decision rules from decision tree classifiers and developed an SSE-based scheme (SDTC) with $\mathcal{O}(1)$ computational complexity and 1 round communication interaction [20]. Yet, since the size of indices in SDTC are exponential to the size of the decision tree classifier, SDTC suffers from prohibitive storage overheads. Furthermore, the large size of indices increase both the computational cost and the communication cost of SDTC. To achieve practical health monitoring services, it is important to design a scheme with low storage costs. Namely, the storage cost should not be exponential to the size of the input domain.

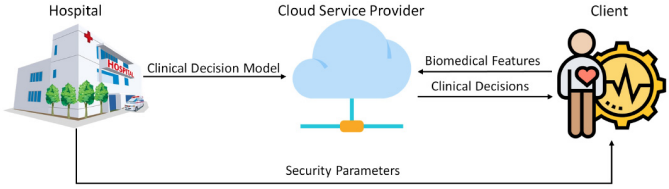


Fig. 1. Health monitoring systems.

Summary: We provide Table I to summarize the differences between the aforementioned schemes. We focus on the same system setting as SDTC [20] and address all the above challenges. To achieve efficient and secure decision tree classification, we transform a decision tree classifier to indices, whose size is polynomial to the size of decision tree classifier. With such indices, the proposed SSE-based scheme is lightweight compared with SDTC in terms of computation, communication, and storage overheads.

III. MODELS AND DESIGN GOALS

A. System and Threat Models

Generally, there are three entities in health monitoring systems, i.e., a hospital ($\mathcal{H}$), a cloud service provider ($\mathcal{CSP}$), and a client ($\mathcal{C}$). As shown in Fig. 1, the procedure of health monitoring systems could be described as follows.

1) *Hospital ($\mathcal{H}$):* $\mathcal{H}$ owns a clinical decision model, which is pretrained by utilizing decision tree classification techniques. By outsourcing the clinical decision model to $\mathcal{CSP}$, $\mathcal{H}$ offers health monitoring services to remote $\mathcal{C}$ via $\mathcal{CSP}$ and, therefore, could keep offline and focus on treating patients in emergency situations.
2) *Client ($\mathcal{C}$):* $\mathcal{C}$ is a patient with chronic diseases that need to be monitored periodically. With wearable sensors, $\mathcal{C}$ regularly tests his/her biomedical features (e.g., heart rate, blood pressure, and blood oxygen concentration). Due to computation resource constraints and network connectivity challenges of wearable devices, it would not be realistic for $\mathcal{C}$ to calculate the clinical decision for his/her biomedical features locally or to participate in the decision-making process actively by always staying online. As a result, $\mathcal{C}$ stays offline after uploading the biomedical features to $\mathcal{CSP}$ and later retrieves the corresponding clinical decision from $\mathcal{CSP}$ by leveraging the cloud-empowered decision tree classification.
3) *Cloud Service Provider ($\mathcal{CSP}$):* $\mathcal{CSP}$ obtains the clinical decision model from $\mathcal{H}$ and provides health monitoring services to $\mathcal{C}$. When $\mathcal{CSP}$ receives the biomedical features from $\mathcal{C}$ periodically, $\mathcal{CSP}$ evaluates the biomedical

features by utilizing the outsourced decision tree model and returns the corresponding clinical decision to $\mathcal{C}$.

In health monitoring systems, both $\mathcal{H}$ and $\mathcal{C}$ are honest entities, which own the clinical decision model and biomedical data (including the biomedical features and the clinical decisions), respectively. Since $\mathcal{CSP}$ is a third-party service provider, $\mathcal{CSP}$ is always viewed as a semihonest entity, which follows the procedure honestly but may be interested in the valuable clinical decision model and the sensitive biomedical data. As a result, the privacy of the clinical decision model and the biomedical data should be preserved against $\mathcal{CSP}$. We assume that both $\mathcal{H}$ and $\mathcal{C}$ will not collude with $\mathcal{CSP}$.

B. Design Goals

In this article, we aim to design an efficient and PPDT classification scheme for health monitoring systems, which should achieve the following properties.

1) *Confidentiality:* The confidentiality goal of PPDT is to protect sensitive data against $\mathcal{CSP}$, which contains two confidentiality requirements as follows.
   a) *Data Confidentiality:* Since biomedical features and clinical decisions are sensitive data for $\mathcal{C}$, the confidentiality of biomedical data should keep secret against $\mathcal{CSP}$.
   b) *Model Confidentiality:* Due to intellectual property protection issues, the clinical decision model is valuable knowledge assets for $\mathcal{H}$. Thus, the confidentiality of the clinical decision model should be protected against $\mathcal{CSP}$.
2) *Efficiency:* The efficiency goal of PPDT is to achieve efficient and real-time health monitoring services, which involves three requirements as follows.
   a) *Low Computational Complexity:* To achieve efficient and PPDT classification, the proposed scheme should achieve sublinear computational complexity and avoid using cryptographic techniques with expensive costs.
   b) *Low Communication Costs:* To achieve real-time health monitoring services, the communication costs of the proposed scheme should be low.
   c) *Low Storage Costs:* To reduce the healthcare costs of deploying the health monitoring system, the storage costs of the proposed scheme should be low.

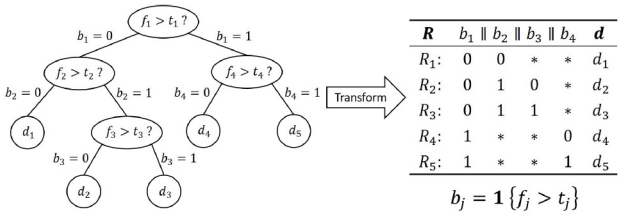
IV. Preliminaries

A. Cryptographic Preliminaries

Pseudorandom Functions: Prf are keyed functions whose outputs are computationally indistinguishable from random values [33].

Definition 1: A keyed function $\{0,1\}^\kappa \times \{0,1\}^t \to \{0,1\}^l$ is a *Prf* if for all probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\left|\Pr\left[\mathcal{A}^{\mathrm{Prf}_k(\cdot)}(1^t)=1\right]-\Pr\left[\mathcal{A}^{\mathrm{Rnd}_t(\cdot)}(1^t)=1\right]\right|\le \mathrm{negl}(t)$$



$\boldsymbol{R}$	$b_1 \| b_2 \| b_3 \| b_4$	$\boldsymbol{d}$
R_1:	0 0 * *	d_1
R_2:	0 1 0 *	d_2
R_3:	0 1 1 *	d_3
R_4:	1 * * 0	d_4
R_5:	1 * * 1	d_5

Fig. 2. Example of decision tree classification.

where the key $k \leftarrow \{0,1\}^\kappa$ is a randomly chosen κ-bit string and Rnd_t is randomly selected from the set of functions mapping t-bit strings to l-bit strings.

Pseudorandom Permutations: Prp are keyed bijections whose output cannot be computationally distinguished from a permutation that is randomly chosen from the set of all permutations on the function's domain [33].

Definition 2: A keyed permutation $\{0,1\}^\kappa \times \{0,1\}^t \to \{0,1\}^t$ is a *Prp* if for all probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\left|\Pr\left[\mathcal{A}^{Prp_k(\cdot)}(1^t)=1\right]-\Pr\left[\mathcal{A}^{\mathrm{Rnd}_t(\cdot)}(1^t)=1\right]\right|\le \mathrm{negl}(t)$$

where the key $k \leftarrow \{0,1\}^\kappa$ is a randomly chosen κ-bit string and Rnd_t is selected uniformly at random from the set of all functions permutating t-bit strings to t-bit strings.

Symmetric Key Encryption: Symmetric key encryption (Sym) denotes any probabilistic encryption, whose encryption key is the same as the decryption key [33]. In this article, we assume Sym achieves the indistinguishability under chosen plaintext attacks (IND-CPAs).

B. Decision Tree Classification and Rule Extraction

Decision tree classification is a well-known machine learning technique used in smart healthcare applications. As shown in Fig. 2, a decision tree contains a set of internal nodes and leaf nodes, which associate with constant thresholds and predictions, respectively. Let $\boldsymbol{t} = \{t_1, t_2, \ldots, t_j, \ldots, t_n\}$ be a set of thresholds associated with all internal nodes, where n denotes the number of thresholds. Let $\boldsymbol{d} = \{d_1, d_2, \ldots, d_i, \ldots, d_m\}$ be a set of predictions associated with all leaf nodes, where m denote the number of predictions. With an n-dimensional feature $\boldsymbol{f} = \{f_1, f_2, \ldots, f_j, \ldots, f_n\}$ as input, the decision tree classification process is described as follows. First, from the root node, the value of feature f_j is compared with the threshold t_j. Second, the comparison result decides which child node (either left child node when $f_j \le t_j$ or right child node when $f_j > t_j$) should be taken for comparison next. These procedures are repeated until a leaf node i, which represents the prediction is d_i, is reached.

Recent works, such as [20], have investigated that several decision rules could be extracted from the decision tree classifier. For example, to extract rule R_1 in Fig. 2, the decision rule extraction method proceeds as follows. First, by traversing the decision tree from the root node to the leaf node d_1, condition $f_1 \le t_1$ and $f_2 \le t_2$ are obtained. For the ease of description, we use a bit $b_j = \mathbf{1}\{f_j > t_j\}$ to denote the condition of comparison result of f_j and t_j, where $j \in \{1, 2, \ldots, 4\}$.

TABLE II
NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
DT	The decision tree classifier trained from biomedical data.
n	The number of internal nodes (a.k.a. thresholds) in DT.
m	The number of leaf nodes (a.k.a. predictions) in DT.
$\boldsymbol{t}$	$\boldsymbol{t} = \{t_1, t_2, \ldots, t_n\}$ are values of thresholds in DT.
$\boldsymbol{d}$	$\boldsymbol{d} = \{d_1, d_2, \ldots, d_m\}$ are values of predictions in DT.
$\boldsymbol{f}$	$\boldsymbol{f} = \{f_1, f_2, \ldots, f_n\}$ is the input n-dimensional biomedical feature vector.
$[w]$	The set of integers $\{1, \ldots, \mathrm{w}\}$.
$\boldsymbol{R}$	$\boldsymbol{R} = \{R_1, \ldots, R_m\}$ are rules that extracted from DT.
R_i	$R_i = \{R_{i,1}, R_{i,2}, \ldots, R_{i,n}, d_i\}$, where $R_{i,j}$ denotes the value of b_j in R_i, d_i denotes the value of prediction of R_i.
$\boldsymbol{V}$	$\boldsymbol{V} = \{\boldsymbol{v}_{1,1}, \ldots, \boldsymbol{v}_{1,n}, \ldots, \boldsymbol{v}_{i,j}, \ldots, \boldsymbol{v}_{m,1}, \ldots, \boldsymbol{v}_{m,n}\}$ be $m \times n$ boolean vectors, where $\boldsymbol{v}_{i,j}$ has w elements.
F_0	The $\kappa + \log \kappa$ bit $\mathtt{prf}$ with a κ bit key.
K_f	The κ bit key of F_0.
H_0	The $\log(mnw)$ bit $\mathtt{prp}$ with a κ bit key.
K_0	The κ bit key of H_0.
H_1	The $\log m$ bit $\mathtt{prp}$ with a κ bit key.
K_1	The κ bit key of H_1.
$\mathtt{Sym}$	The IND-CPA secure symmetric key encryption.
K_d	A symmetric key produced by $\mathtt{Sym}$.
K	A symmetric key produced by $\mathtt{Sym}$.
ER	The linear index with mnw elements.
EP	The linear index with m elements.
$\boldsymbol{TK}$	$\boldsymbol{TK} = \{TK_1, \ldots, TK_i, \ldots, TK_m\}$ are m tokens.
TK_i	$TK_i = (TK_i^1, TK_i^2, TK_i^3, \boldsymbol{L}_i)$.
$\boldsymbol{L}_i$	n different locations in ER.

Namely, $b_1 = 0$ and $b_2 = 0$ are obtained in the first step. Second, since the conditions b_3 and b_4 are not included in the decision path from the root node to d_1, both b_3 and b_4 are set to be $*$, which is a wildcard ("don't care" value). Third, all Boolean conditions are concatenated into $b_1 \| b_2 \| b_3 \| b_4$, which is $00**$ for the path from the root node to d_1. As a result, R_1 is extracted, which includes the path $00**$ and the prediction d_1. By repeating the above method from the root node to each leaf node, five decision rules can be extracted from the decision tree in Fig. 2, i.e., $\boldsymbol{R} = \{R_1, \ldots, R_5\}$.

With the extracted rules $\boldsymbol{R}$, the process of decision tree classification can be transformed into two phases: 1) internal node comparison and 2) decision path evaluation. In the internal node comparison phase, a comparison result c_j is produced after comparing each feature $f_j \in \boldsymbol{f}$ with a threshold $t_j \in \boldsymbol{t}$. In the decision path evaluation phase, all comparison results are concatenated as $c_1 \| \cdots \| c_n$. When the concatenation of comparison result $c_1 \| \cdots \| c_n$ matches the decision path (i.e., $b_1 \| \cdots \| b_n$) in R_i, the corresponding prediction for $\boldsymbol{f}$ is d_i, where $d_i \in \boldsymbol{d}$.

V. DESIGN OF PPDT

A. Definitions

Let DT be a decision tree classifier with n internal nodes (thresholds) and m leaf nodes (predictions), which is trained from biomedical data. Let $\boldsymbol{t} = \{t_1, \ldots, t_n\}$ and $\boldsymbol{d} = \{d_1, \ldots, d_m\}$ be the sets of thresholds and predictions, respectively. Let $\boldsymbol{f} = \{f_1, \ldots, f_n\}$ be an n-dimensional biomedical feature vector. Let $[w]$ be a set of integers $\{1, 2, \ldots, w\}$. Without loss of generality, we assume that all biomedical features and thresholds in decision trees are all positive integers, and the domains of them are $[w]$, i.e., $\boldsymbol{f}, \boldsymbol{t} \in [w]^n$. Let $\boldsymbol{R} = \{R_1, R_2, \ldots, R_m\}$ be m decision rules extracted from DT with m leaf nodes. Each rule $R_i \in \boldsymbol{R}$ is expressed as $R_i = \{R_{i,1}, \ldots, R_{i,j}, \ldots, R_{i,n}, d_i\}$, where $R_{i,j}$ denotes the value of b_j in the rule R_i. Namely, 0, 1, and the wildcard $*$ are potential values for $R_{i,j}$, because $R_{i,j}$ denotes the relationship between f_j and t_j in the rule R_i. Let $\boldsymbol{V} = \{\boldsymbol{v}_{1,1}, \ldots, \boldsymbol{v}_{1,n}, \ldots, \boldsymbol{v}_{i,j}, \ldots, \boldsymbol{v}_{m,1}, \ldots, \boldsymbol{v}_{m,n}\}$ be $m \times n$ Boolean vectors, which represent all decision paths in $\boldsymbol{R}$. Each $\boldsymbol{v}_{i,j} \in \boldsymbol{V}$ denotes $R_{i,j} \in R_i$ and has w elements.

Let κ be the security parameter. Let $F_0 : \{0,1\}^\kappa \times \{0,1\}^{\kappa + \log \kappa} \to \{0,1\}^{\kappa + \log \kappa}$ be a $\mathtt{prf}$, where $K_f \leftarrow \{0,1\}^\kappa$ is the key of F_0. Let $H_0 : \{0,1\}^\kappa \times \{0,1\}^{\log(mnw)} \to \{0,1\}^{\log(mnw)}$ and $H_1 : \{0,1\}^\kappa \times \{0,1\}^{\log m} \to \{0,1\}^{\log m}$ be two $\mathtt{prp}$ functions, where $K_0 \leftarrow \{0,1\}^\kappa$ and $K_1 \leftarrow \{0,1\}^\kappa$ are the keys of H_0 and H_1, respectively. Let $\mathtt{Sym} = (\mathtt{Sym.Gen}, \mathtt{Sym.Enc}, \mathtt{Sym.Dec})$ be an IND-CPA secure symmetric key encryption, whose key space and plaintext space are $\{0,1\}^{\kappa + \log \kappa}$. Let K_d and K be two symmetric keys of $\mathtt{Sym}$. Let ER be a linear index for encrypted decision rules extracted from DT, which has mnw elements. Let EP be a linear index for encrypted predictions in DT, which has m elements. $\boldsymbol{TK} = \{TK_1, \ldots, TK_m\}$ are m tokens for achieving the clinical decisions. Each $TK_i \in \boldsymbol{TK}$ can be represented as $TK_i = (TK_i^1, TK_i^2, TK_i^3, \boldsymbol{L}_i)$, where $\boldsymbol{L}_i$ denotes n locations in ER. All notations are summarized in Table II.

B. Boolean Vectors and Decision Trees

The basic idea of this article is to transform the DT to $m \times n$ Boolean vectors $\boldsymbol{V}$ and utilize the encrypted $\boldsymbol{V}$ for achieving PPDT classification. Now, we describe how to transform the decision tree classifier DT into Boolean vectors $\boldsymbol{V}$ as follows.

After DT is trained from biomedical data, $\mathcal{H}$ extracts m rules from DT, i.e., $\boldsymbol{R} = \{R_1, \ldots, R_m\}$. For each rule $R_i \in \boldsymbol{R}$, $\mathcal{H}$ builds n Boolean vectors, i.e., $\boldsymbol{v}_{i,1}, \ldots, \boldsymbol{v}_{i,n}$. Each Boolean vector $\boldsymbol{v}_{i,j} \in \boldsymbol{V}$, which has w elements, is used for representing $R_{i,j} \in R_i$. Then, the value of each element $v_{i,j}[k]$ in $\boldsymbol{v}_{i,j}$ is set as in

$$v_{i,j}[k] \leftarrow \begin{cases} 1, & \text{if } R_{i,j} = 1 \text{ and } k > t_j \\ 1, & \text{if } R_{i,j} = 0 \text{ and } k \le t_j \\ 1, & \text{if } R_{i,j} = * \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

With the biomedical feature vector $\boldsymbol{f}$ as input, the decision tree classification process in $\boldsymbol{V}$ proceeds as follows. For each rule R_i, $\mathcal{C}$ tests n values in $\boldsymbol{V}$, i.e., $v_{i,1}[f_1], v_{i,2}[f_2], \ldots, v_{i,n}[f_n]$. If these values are all equal to 1, then the prediction of $\boldsymbol{f}$ is d_i. Namely, for rule R_i, if $\forall v_{i,j}[f_j] = 1$, where $j \in [n]$, then the prediction of $\boldsymbol{f}$ is d_i.

For example, the value of R_2 in Fig. 2 is $\{0, 1, 0, *\}$. Assume that $w = 10$ and $\boldsymbol{t} = \{4, 6, 3, 7\}$, then the value of the Boolean vectors $\boldsymbol{v}_{2,1}, \boldsymbol{v}_{2,2}, \boldsymbol{v}_{2,3}$, and $\boldsymbol{v}_{2,4}$ are shown in Fig. 3. For the Boolean vector $\boldsymbol{v}_{2,1}$, since $t_1 = 4$ and $R_{2,1} = 0$, elements from $v_{2,1}[1]$ to $v_{2,1}[4]$ are set to be 1, while elements from $v_{2,1}[5]$

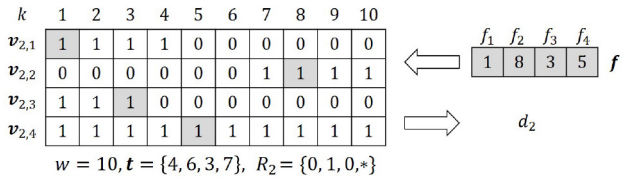


Fig. 3. Example of the Boolean vectors generated from rule R_2.

to $v_{2,1}[10]$ are set to be 0. For the Boolean vector $\boldsymbol{v}_{2,2}$, since $t_2 = 6$ and $R_{2,2} = 1$, elements from $v_{2,2}[1]$ to $v_{2,2}[6]$ are set to be 0, while elements from $v_{2,2}[7]$ to $v_{2,2}[10]$ are set to be 1. For the Boolean vector $\boldsymbol{v}_{2,3}$, since $t_3 = 3$ and $R_{2,3} = 0$, elements from $v_{2,3}[1]$ to $v_{2,3}[3]$ are set to be 1, while elements from $v_{2,3}[4]$ to $v_{2,3}[10]$ are set to be 0. For the Boolean vector $\boldsymbol{v}_{2,4}$, since $t_4 = 7$ and $R_{2,3} = *$, elements from $v_{2,3}[1]$ to $v_{2,3}[10]$ are set to be 1. With the feature vector $\boldsymbol{f} = \{1, 8, 3, 5\}$, since both the values of $v_{2,1}[1]$, $v_{2,2}[8]$, $v_{2,3}[3]$, and $v_{2,4}[5]$ are equal to 1, the feature vector $\boldsymbol{f}$ matches R_2 and, therefore, the corresponding prediction of $\boldsymbol{f}$ is d_2.

C. Detailed Design of the PPDT Scheme

The PPDT classification scheme for health monitoring systems is defined as follows.

Definition 3 (PPDT Classification): The *PPDT* involves four polynomial-time algorithms, i.e., *PPDT = (Init, ClfEnc, TokenGen, Eva)*.

1) *Init* $(\kappa) \rightarrow K_f, K_0, K_1, H_0, H_1, F_0, K_d$. The initialization algorithm is run by $\mathcal{H}$. Based on the security parameter κ, $\mathcal{H}$ produces $K_f, K_0, K_1, H_0, H_1, F_0$, and K_d and sends K_f, K_0, K_1, H_0, H_1, F_0, and K_d to authorized $\mathcal{C}$.
2) *ClfEnc* $(K_f, K_0, K_1, H_0, H_1, F_0, K_d, \boldsymbol{V}, \boldsymbol{d}) \rightarrow ER, ED$. The classifier encryption algorithm is run by $\mathcal{H}$. First, for each value in $\boldsymbol{V}$, $\mathcal{H}$ encrypts $\boldsymbol{V}$ and stores the encrypted values to the linear index ER. Then, for each prediction d_i in $\boldsymbol{d}$, $\mathcal{H}$ encrypts d_i, and stores the encrypted values to the encrypted linear index EP. Finally, $\mathcal{H}$ outsources ER and EP to $\mathcal{CSP}$.
3) *TokenGen* $(K_f, K_0, K_1, H_0, H_1, F_0, \boldsymbol{f}) \rightarrow \boldsymbol{TK}$. The token generation algorithm is run by $\mathcal{C}$. When $\mathcal{C}$ requires a clinical decision for his/her biomedical features $\boldsymbol{f}$, he/she produces a set of tokens $\boldsymbol{TK}$ for $\boldsymbol{f}$ and outsources the $\boldsymbol{TK}$ to $\mathcal{CSP}$.
4) *Eva* $(\boldsymbol{TK}, ER, ED, K_d) \rightarrow d_i$. The evaluation algorithm is an interactive algorithm run by $\mathcal{CSP}$ and $\mathcal{C}$. After receiving $\boldsymbol{TK}$ from $\mathcal{C}$, $\mathcal{CSP}$ produces a clinical decision by searching ER and EP, and returns the encrypted prediction to $\mathcal{C}$. Then, $\mathcal{C}$ decrypts the encrypted prediction and receives d_i as the evaluation result for $\boldsymbol{f}$, where $i = 0, 1, \ldots, t$.

Scheme Details: The main idea of PPDT could be described as follows. First, we transform a DT to $m \times n$ Boolean vectors $\boldsymbol{V}$ and extract all predictions $\boldsymbol{d}$ in DT, where d_i is the corresponding prediction for Boolean vectors $\boldsymbol{v}_{i,1}, \ldots, \boldsymbol{v}_{i,n}$. Second, we utilize the scheme in [34] and Sym to encrypt $\boldsymbol{V}$ and $\boldsymbol{d}$, respectively. Third, we construct two indices for DT by permutating the encrypted $\boldsymbol{V}$ and $\boldsymbol{d}$ with prp. Finally, PPDT classification could be achieved by searching the encrypted indices. We show the detail construction of PPDT in Fig. 4.

With PPDT, the workflow of privacy-preserving health monitoring systems contains two processes, i.e., the setup process and the decision process. In the setup process, $\mathcal{H}$ outsources the encrypted clinical decision model to $\mathcal{CSP}$ and shares several parameters to $\mathcal{C}$ by utilizing PPDT. First, $\mathcal{H}$ invokes the PPDT.Init to initialize several parameters and share these parameters to authorized $\mathcal{C}$. Then, $\mathcal{H}$ invokes the PPDT.ClfEnc to encrypt the Boolean vectors $\boldsymbol{V}$ and the medical prediction $\boldsymbol{d}$. The encrypted $\boldsymbol{V}$ and $\boldsymbol{d}$ are pseudorandomly stored in ER and EP, respectively. After that, $\mathcal{H}$ outsources ER and EP to $\mathcal{CSP}$ for offering health monitoring services to remote $\mathcal{C}$.

In the decision process, $\mathcal{C}$ submits his/her biomedical features to $\mathcal{CSP}$ periodically and receives the clinical decision from $\mathcal{CSP}$. First, $\mathcal{C}$ invokes the PPDT.TokenGen algorithm to generate tokens $\boldsymbol{TK}$ for his/her biomedical features $\boldsymbol{f}$ and submits $\boldsymbol{TK}$ to $\mathcal{CSP}$. Then, $\mathcal{CSP}$ and $\mathcal{C}$ engage in the PPDT.Eva protocol. Namely, $\mathcal{CSP}$ returns the corresponding encrypted clinical decision by searching ER and EP, while $\mathcal{C}$ decrypts the encrypted clinical decision and obtains the prediction for $\boldsymbol{f}$.

Correctness: The correctness of PPDT could be verified as follows. Assume that the prediction of $\boldsymbol{f}$ is d_l, where $l \in [m]$. We consider the following two cases for the produced token $\boldsymbol{TK} = \{TK_1, \ldots, TK_i, \ldots, TK_m\}$ as follows.

Case 1: If $i = l$, we will have $v_{i,j}[f_j] = 1$, where $j \in [n]$, because $\boldsymbol{f}$ matches R_l. Then, we will have $F_0(K_f, v_{i,j}[f_j] \| i \| j \| f_j) = F_0(K_f, 1 \| i \| j \| f_j))$. We can find that

$$\begin{aligned} K' &= \oplus_{f_j \in \boldsymbol{f}} \left(ER\left[H_0\left(K_0, i \| j \| f_j \right) \right] \right) \oplus TK_i^1 \\ &= \oplus_{f_j \in \boldsymbol{f}} \left(F_0\left(K_f, v_{i,j}[f_j] \| i \| j \| f_j \right) \right) \oplus TK_i^1 \\ &= \oplus_{f_j \in \boldsymbol{f}} \left(F_0\left(K_f, 1 \| i \| j \| f_j \right) \right) \oplus TK_i^1 \\ &= K. \end{aligned}$$

Thus, we find that

$$\texttt{Sym.Dec}\left(K', TK_i^2 \right) = 0^{\kappa + \log \kappa}$$

and

$$\texttt{Sym.Dec}\left(K', TK_i^3 \right) = H_1(K_1, i).$$

Therefore, the corresponding prediction is

$$\texttt{Sym.Dec}(K_d, EP[H_1(K_1, i)]) = d_i = d_l.$$

Case 2: If $i \neq l$, we must have $v_{i,j}[f_j] \neq 1$, for some $f_j \in \boldsymbol{f}$, because some features $f_j \in \boldsymbol{f}$ cannot satisfy $f_j > t_j$. This, in turn, implies that for some $j \in [n]$, $F_0(K_f, v_{i,j}[f_j] \| i \| j \| f_j) \neq F_0(K_f, 1 \| i \| j \| f_j))$. Thus, after the bitwise XOR operation, $K' \neq K$. This ensures that with all but negligible probability, $\texttt{Sym.Dec}(K', TK_i^2) \neq 0^{\kappa + \log \kappa}$, which means $\boldsymbol{f}$ does not match R_i. Thus, TK_i cannot be used for obtaining d_i.

VI. Security Analysis

A. Leakage Functions and Security Definition

Before formulating the security definition of PPDT, we first define a leakage function $\mathcal{L}$ for PPDT, which describes

Privacy-Preserving Decision Tree Classification (PPDT)

★ **Initialization**: PPDT.Init $(\kappa) \rightarrow K_f, K_0, K_1, H_0, H_1, F_0, K_d$.

1: $\mathcal{H}$: $\mathcal{H}$ chooses a security parameter κ and samples K_f from $\{0,1\}^\kappa$, i.e.,

$$K_f \xleftarrow{\$} \{0,1\}^\kappa.$$

Then, $\mathcal{H}$ produces a symmetric keys K_d for $\mathtt{Sym}$, i.e.,

$$K_d \leftarrow \mathtt{Sym.Gen}\left(1^\kappa\right).$$

After that, $\mathcal{H}$ randomly produces two pseudo-random permutation H_0 and H_1, where

$$H_0 : \{0,1\}^\kappa \times \{0,1\}^{\log(mnw)} \rightarrow \{0,1\}^{\log(mnw)},$$
$$H_1 : \{0,1\}^\kappa \times \{0,1\}^{\log(m)} \rightarrow \{0,1\}^{\log(m)}.$$

$\mathcal{H}$ samples two keys K_0 and K_1 for H_0 and H_1, respectively. That is,

$$K_0 \xleftarrow{\$} \{0,1\}^\kappa, \quad K_1 \xleftarrow{\$} \{0,1\}^\kappa.$$

2: $\mathcal{H} \rightarrow \mathcal{C}$: $\mathcal{H}$ sends K_f, K_0, K_1, K_d, H_0, H_1, and F_0 to authorized $\mathcal{C}$.

★ **Classifier Encryption**: PPDT.ClfEnc $(K_f, K_0, K_1, H_0, H_1, F_0, K_d, \boldsymbol{V}, \boldsymbol{d}) \rightarrow ER, EP$.

1: $\mathcal{H}$: For $i \in [m]$, $j \in [n]$, $k \in [w]$, $\mathcal{H}$ sets:

$$ER\left[H_0\left(K_0, i||j||k\right)\right] \leftarrow F_0\left(K_f, v_{i,j}[k]||i||j||k\right).$$

2: $\mathcal{H}$: For $i \in [m]$, $\mathcal{H}$ sets:

$$EP[H_1(K_1, i)] \leftarrow \mathtt{Sym.Enc}(K_d, d_i),$$

where $d_i \in \boldsymbol{d}$.

3: $\mathcal{H} \rightarrow \mathcal{CSP}$: $\mathcal{H}$ outsources ER and EP to $\mathcal{CSP}$.

★ **Token Generation**: PPDT.TokenGen $(K_f, K_0, H_0, F_0, \boldsymbol{f}) \rightarrow \boldsymbol{TK}$.

1: $\mathcal{C}$: When $\mathcal{C}$ requests a clinical decision for his/her biomedical features $\boldsymbol{f} = \{f_1, f_2, \ldots, f_n\}$, he/she samples a key K for $\mathtt{Sym}$, i.e.,

$$K \xleftarrow{\$} \{0,1\}^\kappa.$$

Then, he/she produces m tokens $\boldsymbol{TK} = \{TK_1, TK_2, \ldots, TK_m\}$. Token TK_i contains three values and a vector, i.e., $TK_i = \left(TK_i^1, TK_i^2, TK_i^3, \boldsymbol{L}_i\right)$, which are calculated as follows.

$$TK_i^1 = \oplus_{f_j \in \boldsymbol{f}} \left(F_0(K_f, 1||i||j||f_j)\right) \oplus K,$$
$$TK_i^2 = \mathtt{Sym.Enc}(K, 0^{\kappa + \log \kappa}),$$
$$TK_i^3 = \mathtt{Sym.Enc}(K, H_1(K_1, i)),$$
$$\boldsymbol{L}_i = \{H_0(K_0, i||j||f_j)\}_{j \in [n]}.$$

2: $\mathcal{C} \rightarrow \mathcal{CSP}$: $\mathcal{C}$ submits $\boldsymbol{TK}$ to $\mathcal{CSP}$.

★ **Evaluation**: PPDT.Eva $(\boldsymbol{TK}, ER, ED, K_d) \rightarrow d_i$.

1: $\mathcal{CSP}$: $\mathcal{CSP}$ receives ER and EP from $\mathcal{H}$.

2: $\mathcal{CSP}$: After receiving $\boldsymbol{TK}$ from $\mathcal{C}$, for $i \in [m]$, $\mathcal{CSP}$ calculates

$$K' = \oplus_{f_j \in \boldsymbol{f}} \left(ER\left[H_0(K_0, i||j||f_j)\right]\right) \oplus TK_i^1.$$

3: $\mathcal{CSP} \rightarrow \mathcal{U}$: If $\mathtt{Sym.Dec}\left(K', TK_i^2\right) = 0^{\kappa + \log \kappa}$, then $\mathcal{CSP}$ searches $EP\left[\mathtt{Sym.Dec}(K', TK_i^3)\right]$ and obtains $\mathtt{Sym.Enc}(K_d, d_i)$, where $i \in [m]$. Then, $\mathcal{CSP}$ returns $\mathtt{Sym.Enc}(K_d, d_i)$ to $\mathcal{U}$.

4: $\mathcal{U}$: After receiving $\mathtt{Sym.Enc}(K_d, d_i)$, $\mathcal{U}$ calculates

$$d_i = \mathtt{Sym.Dec}(K_d, \mathtt{Sym.Enc}(K_d, d_i)),$$

where $i \in [m]$. Finally, d_i is the corresponding clinical decision for $\boldsymbol{f}$.

Fig. 4. Detailed construction of PPDT for health monitoring systems.

the information revealed when processing PPDT classification. The inputs of PPDT are the Boolean vector $\boldsymbol{V}$ and the biomedical feature vector $\boldsymbol{f}$. $\mathcal{L}(\boldsymbol{V}, \boldsymbol{f})$ is defined as follows.

Definition 4: $\mathcal{L}(\boldsymbol{V}, \boldsymbol{f})$. The leakage function $\mathcal{L}$ involves size pattern, access pattern, and search pattern.

1) *Size Pattern:* The size pattern involves the number of elements for index *ER* and *EP*, and the number of tokens that have been submitted. Namely, the size pattern denotes the size of both indices and tokens.
2) *Access Pattern:* The access pattern is the mapping relation between the submitted tokens and the corresponding encrypted prediction. Namely, the access pattern denotes how to retrieve the prediction for a token.
3) *Search Pattern:* The search pattern is the differences between the two input tokens. Namely, the search pattern indicates whether a token has been searched.

The leaked information in $\mathcal{L}(\boldsymbol{V}, \boldsymbol{f})$ is usually considered default leaked information in most of the searchable encryption schemes and secure decision tree classification scheme. With the leakage function $\mathcal{L}(\boldsymbol{V}, \boldsymbol{f})$, the adaptive $\mathcal{L}$-security definition can be formulated.

Definition 5: Adaptive $\mathcal{L}$-security. Let $\Phi = (\mathit{Init}, \mathit{ClfEnc}, \mathit{TokenGen}, \mathit{Eva})$ be a PPDT classification scheme. Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_q)$ and $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_q)$ be an adversary and a simulator, respectively, where $q \in \mathbb{N}$. Let $\boldsymbol{f}^1, \boldsymbol{f}^2, \ldots, \boldsymbol{f}^q$ be biomedical feature vectors for q PPDT classification requests generated by $\mathcal{A}$. We define the $\mathit{Real}_\Phi^\mathcal{A}(1^\kappa)$ experiment and the $\mathit{Sim}_{\mathcal{L},\mathcal{S}}^\mathcal{A}(1^\kappa)$ experiment as follows.

$\mathit{Real}_\Phi^\mathcal{A}(1^\kappa)$*:* At round 0, the challenger invokes $\mathit{Init}(\kappa)$ to produce K_f, K_0, K_1, H_0, H_1, F_0, and K_d. Then, $\mathcal{A}_0$ produces a decision tree DT and extracts Boolean

vectors $\boldsymbol{V}$ and predictions $\boldsymbol{d}$ from DT. The challenger invokes $ClfEnc(K_f, K_0, K_1, H_0, H_1, F_0, K_d, \boldsymbol{V}, \boldsymbol{d})$ to generate encrypted indices ER and EP, and sends ER and EP to $\mathcal{A}$. After that, $\mathcal{A}$ makes q classification requests. At round r ($1 \le r \le q$): $\mathcal{A}_r$ reviews the previous requests and generates $\boldsymbol{f}^r$ adaptively. The challenger invokes $TokenGen(K_f, K_0, H_0, F_0, \boldsymbol{f})$ to produce $\boldsymbol{TK}^r$, and submits $\boldsymbol{TK}^r$ to $\mathcal{A}_r$. Then, $\mathcal{A}_r$ searches the encrypted indices ER and EP by utilizing $\boldsymbol{TK}^r$ and obtains an encrypted prediction for $\boldsymbol{f}^r$. After q round interactions, $\mathcal{A}$ produces a bit as the output.

$\boldsymbol{Sim}^{\mathcal{A}}_{\mathcal{L},\mathcal{S}}(1^\kappa)$: At round 0, $\mathcal{S}_0$ randomly generates two indices ER^* and EP^* by utilizing $\mathcal{L}(\boldsymbol{V},\boldsymbol{f})$ and sends both ER^* and EP^* to $\mathcal{A}$. Then, $\mathcal{A}$ makes q classification requests. At round r ($1 \le r \le q$): $\mathcal{A}_r$ reviews the previous requests and generates $\boldsymbol{f}^r$ adaptively. With $\mathcal{L}(\boldsymbol{V},\boldsymbol{f})$, $\mathcal{S}_r$ generates m appropriate tokens $\boldsymbol{TK}^{r*} = \{TK_1^{r*}, \ldots, TK_i^{r*}, \ldots, TK_m^{r*}\}$, where $TK_i^{r*} = (TK_i^{1r*}, TK_i^{2r*}, TK_i^{3r*}, \boldsymbol{L}_i)$. After that, $\mathcal{S}_r$ submits $\boldsymbol{TK}^{r*}$ to $\mathcal{A}_r$. Then, $\mathcal{A}_r$ searches the encrypted indices ER^* and EP^* by utilizing $\boldsymbol{TK}^{r*}$, and obtains an encrypted prediction for $\boldsymbol{f}^r$. After q round interactions, $\mathcal{A}$ produces a bit as the output.

We say that Φ is adaptively $\mathcal{L}$-secure if for all polynomial size adversaries $\mathcal{A} = (\mathcal{A}_0, \ldots, \mathcal{A}_q)$, there exist a simulator $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_q)$ and a negligible function $\mathrm{negl}(\kappa)$ such that

$$\left|\Pr\left[\boldsymbol{Real}^{\mathcal{A}}_{\Phi}(1^\kappa) = 1\right] - \Pr\left[\boldsymbol{Sim}^{\mathcal{A}}_{\mathcal{L},\mathcal{S}}(1^\kappa) = 1\right]\right| \le \mathrm{negl}(\kappa).$$

B. Security Proofs

Theorem 1: $PPDT$ is adaptively $\mathcal{L}$-secure if H_0 and H_1 are Prps, F_0 is a Prf, and Sym is a IND-CPA secure symmetric key encryption.

Proof: We create a simulator $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_q)$ such that for an adversary $\mathcal{A} = (\mathcal{A}_0, \ldots, \mathcal{A}_q)$, the outputs of $\mathbf{Real}^{\mathcal{A}}_{\Phi}(1^\kappa)$ and $\mathbf{Sim}^{\mathcal{A}}_{\mathcal{L},\mathcal{S}}(1^\kappa)$ are computationally indistinguishable. The simulator $\mathcal{S} = (\mathcal{S}_0, \ldots, \mathcal{S}_q)$ that adaptively produces ER^*, EP^*, and $\boldsymbol{TK}^{r*}$ is constructed as follows.

$\mathcal{S}_0$: With the size pattern in leakage function $\mathcal{L}(\boldsymbol{V},\boldsymbol{f})$, $\mathcal{S}_0$ obtains m, n, and w. To produce ER^*, $\mathcal{S}_0$ constructs a linear index with mnw elements. For each element in ER^*, $\mathcal{S}_0$ sets the value as a random string $\{0,1\}^{\kappa+\log\kappa}$. Namely, each random value in ER^* has the same length as the output of F_0. To produce EP^*, $\mathcal{S}_0$ constructs a linear index with m elements. For each element in EP^*, $\mathcal{S}_0$ sets the value as a random string $\{0,1\}^{\kappa+\log\kappa}$. Namely, each random value in EP^* has the same length as the output of $\mathtt{Sym.Enc}$. Finally, $\mathcal{S}_0$ sends both ER^* and EP^* to $\mathcal{A}_0$. With all but negligible probability, $\mathcal{A}$ cannot obtain K_f and, thus, $\mathcal{A}$ cannot distinguish the pseudorandom output of $F_0(K_f, 1\|i\|j\|k)$ in ER from a $\{0,1\}^{\kappa+\log\kappa}$ bit random string in ER^*, where $i \in [m]$, $j \in [n]$, and $k \in [w]$. Meanwhile, with all but negligible probability, $\mathcal{A}$ cannot obtain K_d and, therefore, $\mathcal{A}$ cannot distinguish the value of $\mathtt{Sym.Enc}(K_d, d_i)$ in EP from a $\{0,1\}^{\kappa+\log\kappa}$ bit random string in ER^*, if $\mathtt{Sym}$ is an IND-CPA secure symmetric encryption. Hence, both ER^* and EP^* are computationally indistinguishable from ER and EP, respectively.

$\mathcal{S}_r$: For $1 \le r \le q$: With the search pattern of the input biomedical feature vector in $\mathcal{L}(\boldsymbol{V},\boldsymbol{f})$, $\mathcal{S}_r$ checks whether the biomedical feature vector $\boldsymbol{f}^r$ has appeared before. There are two different cases.

a) The biomedical feature vector $\boldsymbol{f}^r$ has totally appeared before. Namely, there exists $\boldsymbol{f}^t$, such that $\{f_j^r = f_j^t | j \in [n], 1 \le t < r\}$. Then, $\mathcal{S}_r$ searches the access patterns in leakage function $\mathcal{L}(\boldsymbol{V},\boldsymbol{f})$, and returns $\boldsymbol{TK}^{r*} = \boldsymbol{TK}^{t*}$ as the appropriate token to $\mathcal{A}_r$ for $\boldsymbol{f}^r$.

b) Some values (or all values) of the biomedical feature vector $\boldsymbol{f}^r$ have not appeared before. The token $\boldsymbol{TK}$ could be generated as follows. First, $\mathcal{S}_r$ randomly chooses an element u in EP^*. Second, token $TK_u^{r*} = (TK_u^{1r*}, TK_u^{2r*}, TK_u^{3r*}, \boldsymbol{L}_u^{r*})$ is generated as follows.

i) With the search pattern in $\mathcal{L}(\boldsymbol{V},\boldsymbol{d})$, $\mathcal{S}_r$ checks whether f_j^r has appeared before. If $f_j^r \in \boldsymbol{f}^r$ has appeared before, it means that there exists a $f_j^t = f_j^r$, where $1 \le t < r$. $\mathcal{S}_r$ selects the same location of f_j^t for token u in ER^* as $L_{u,j}^{r*}$ for f_j^r. Otherwise, $\mathcal{S}_r$ randomly selects a location that have not been selected in ER^* as $L_{u,j}^{r*}$. Finally, $\mathcal{S}_r$ produces $\boldsymbol{L}_u^{r*} = \{L_{u,1}^{r*}, \ldots, L_{u,j}^{r*}, \ldots, L_{u,n}^{r*}\}$.

ii) $\mathcal{S}_r$ randomly samples a $\mathtt{Sym}$ key K^*. Then, with $\boldsymbol{L}_u^{r*} = \{L_{u,1}^{r*}, \ldots, L_{u,j}^{r*}, \ldots, L_{u,n}^{r*}\}$, $\mathcal{S}_r$ calculates

$$TK_u^{1r*} = \oplus_{L_{u,j}^{r*} \in \boldsymbol{L}_u^{r*}} \left(ER^*\left[L_{u,j}^{r*}\right]\right) \oplus K^*.$$

iii) $\mathcal{S}_r$ calculates

$$TK_u^{2r*} = \mathtt{Sym.Enc}\left(K^*, 0^{\kappa+\log\kappa}\right).$$

iv) $\mathcal{S}_r$ calculates

$$TK_u^{3r*} = \mathtt{Sym.Enc}\left(K^*, u\right).$$

Third, token $TK_i^{r*} = (TK_i^{1r*}, TK_i^{2r*}, TK_i^{3r*}, \boldsymbol{L}_i^{r*})$, where $i \ne u$, is generated as follows.

i) With the search pattern in $\mathcal{L}(\boldsymbol{V},\boldsymbol{d})$, $\mathcal{S}_r$ checks whether f_j^r has appeared in TK_i^* before. If $f_j^r \in \boldsymbol{f}^r$ has appeared before, it means that there exists a $f_j^t = f_j^r$, where $1 \le t < r$. $\mathcal{S}_r$ selects the same location of f_j^t for token i in ER^* as $L_{i,j}^{r*}$ for f_j^r. Otherwise, $\mathcal{S}_r$ randomly selects a location that have not been selected in ER^* as $L_{i,j}^{r*}$. Finally, $\mathcal{S}_r$ produces $\boldsymbol{L}_i^{r*} = \{L_{i,1}^{r*}, \ldots, L_{i,j}^{r*}, \ldots, L_{i,n}^{r*}\}$.

ii) $\mathcal{S}_r$ selects a $\kappa + \log\kappa$ bit string as TK_i^{1r*}. Namely, $TK_i^{1r*} \xleftarrow{\$} \{0,1\}^{\kappa+\log\kappa}$.

iii) $\mathcal{S}_r$ selects a $\kappa + \log\kappa$ bit string as TK_i^{2r*}. Namely, $TK_i^{2r*} \xleftarrow{\$} \{0,1\}^{\kappa+\log\kappa}$.

iv) $\mathcal{S}_r$ selects a $\kappa + \log\kappa$ bit string as TK_i^{3r*}. Namely, $TK_i^{3r*} \xleftarrow{\$} \{0,1\}^{\kappa+\log\kappa}$.

Then, $\mathcal{S}_r$ submits $\boldsymbol{TK}^{r*} = \{TK_1^{r*}, \ldots, TK_u^{r*}, \ldots, TK_m^{r*}\}$ to $\mathcal{A}_r$. $\mathcal{A}_r$ produces a prediction for $\boldsymbol{f}^r$ by using $\boldsymbol{TK}^{r*}$. With all but negligible probability, $\mathcal{A}_r$ cannot obtain K_0. Hence, $\mathcal{A}_r$ cannot distinguish $\boldsymbol{L}_i^{r*}$ from $\boldsymbol{L}_i^r$ because it is hard to distinguish the output of $\mathtt{prp}$ from the randomly

TABLE III
PARAMETERS FOR PERFORMANCE ANALYSIS

Notation	Meaning
m	The number of leaf nodes of DT.
n	The number of internal nodes of DT.
w	The domain of biomedical features.
$T_{\mathtt{gen}}$	Computational cost of generating a Sym key.
$T_{\mathtt{enc}}$	Computational cost of Sym encryption.
$T_{\mathtt{dec}}$	Computational cost of Sym decryption.
$T'_{\mathtt{prf}}$	Computational cost of generating a prf key.
$T_{\mathtt{prf}}$	Computational cost of calculating a prf.
$T'_{\mathtt{prp}}$	Computational cost of generating a prp key.
$T_{\mathtt{prp}}$	Computational cost of calculating a prp.
$T_{\mathtt{XOR}}$	Computational cost of performing an exclusive-or operation.
$S_{\mathtt{Sym}}$	Size of Sym ciphertexts.
$S_{\mathtt{prf}}$	Size of prf outputs.
$S_{\mathtt{prp}}$	Size of prp outputs.

selected permutation. Similarly, since $\mathcal{A}_r$ cannot obtain K_f, $\mathcal{A}_r$ can distinguish TK_i^{1r*} from TK_i^{1r} with negligible probability under the assumption that distinguishing the output of random string from that of prf is hard. Since Sym is IND-CPA secure symmetric key encryption, both TK_u^{2r*} and TK_i^{2r*} are indistinguishable from TK_u^{2r} and TK_i^{2r}, respectively, where $i \neq u$, because TK_u^{2r*} is encrypted by using different keys from TK_u^{2r} and TK_i^{2r*} is random strings. Similarly, both TK_u^{3r*} and TK_i^{3r*} are indistinguishable from TK_u^{3r} and TK_i^{3r}, respectively, where $i \neq u$. Thus, with all but negligible probability, $\mathcal{A}_r$ cannot distinguish $\boldsymbol{TK}^r$ from $\boldsymbol{TK}^{r*}$. Meanwhile, since $\mathcal{A}_r$ cannot obtain k_d, the encrypted prediction is indistinguishable from the encrypted prediction in real-world game if Sym is an IND-CPA secure symmetric key encryption.

Therefore, $\mathcal{A}$ cannot distinguish the output of $\mathbf{Sim}^{\mathcal{A}}_{\mathcal{L},\mathcal{S}}(1^\kappa)$ from $\mathbf{Real}^{\mathcal{A}}_{\Phi}(1^\kappa)$. ■

In summary, both the clinical decision model and the biomedical data are well protected under the $\mathcal{L}$-security definition. Although the size pattern, the access pattern, and the search pattern are leaked, the content of clinical decision model and the medical data are well protected, because the confidentiality of both the clinical decision model and the medical data are guaranteed by IND-CPA secure symmetric key encryption. To further enhance the security property, re-encryption could be utilized to enable a stronger protection for such patterns by resetting the leakage function [35], [36].

VII. PERFORMANCE ANALYSIS AND EVALUATIONS

A. Performance Analysis

The performance of PPDT depends on several parameters. We provide a list of parameters that is needed for performance analysis in Table III. Note that once a decision tree DT is trained, m, n, and w are constants.

In Table IV, we summarize and compare the theoretical performance properties of PPDT with the privacy-preserving decision tree classification scheme in [20] (SDTC) in terms of the computational costs of algorithms, including Init, ClfEnc, TokenGen, and Eva, the size of both indices and tokens.

Computational Costs: Since all parameters in Table IV are constants, the computation complexity of both PPDT and the scheme in [20] (SDTC) is $\mathcal{O}(1)$. As shown in Table IV, the computational costs of the initialization algorithm of both PPDT and SDTC are the same. Meanwhile, Table IV demonstrates that the computational costs of the classifier encryption algorithm of SDTC is $(w^n) \cdot (2 \cdot T_{\mathtt{prp}} + T_{\mathtt{prf}} + T_{\mathtt{enc}} + T_{\mathtt{XOR}})$, which is an exponential computational cost. By constructing different indices for the decision tree classifier, the computational cost of the classifier encryption algorithm of PPDT is $m \cdot n \cdot w \cdot (T_{\mathtt{prf}} + T_{\mathtt{prp}}) + m \cdot (T_{\mathtt{enc}} + T_{\mathtt{prp}})$, which is significantly boosted to polynomial costs. Also, the computational overheads of the token generation algorithm and evaluation algorithm of PPDT are $T_{\mathtt{gen}} + m \cdot ((n+1)T_{\mathtt{XOR}} + n \cdot (T_{\mathtt{prf}} + T_{\mathtt{prp}}) + 2 \cdot T_{\mathtt{enc}})$ and $m \cdot (n+1) \cdot T_{\mathtt{XOR}} + (\mathcal{O}(n)+2) \cdot T_{\mathtt{dec}}$, which can provide efficient computation by utilizing low-complexity Sym, prf, and prp. Therefore, PPDT not only significantly improves the computational cost of classifier encryption from exponential to polynomial but also achieves efficient computation and makes the health monitoring services more practical.

Storage and Communication Costs: We also provide the theoretical storage and communication costs in Table IV by analyzing the size of indices and tokens. Since the encrypted indices are required to submit by $\mathcal{H}$ and stored at $\mathcal{CSP}$, the size of indices denotes both the communication costs of $\mathcal{H}$ and the storage costs of $\mathcal{CSP}$. SDTC requires $2 \cdot (w^n) \cdot (S_{\mathtt{Sym}} + S_{\mathtt{prf}})$ to store the indices, and PPDT only requires $m \cdot n \cdot w \cdot (S_{\mathtt{prf}}) + m \cdot S_{\mathtt{Sym}}$ to store the indices. Thus, PPDT boosts the exponential storage efficiency to polynomial storage efficiency. The size of tokens denotes the communication costs of $\mathcal{C}$, which sends encrypted biomedical features to $\mathcal{CSP}$. Table IV shows that the size of tokens of PPDT is $m \cdot (S_{\mathtt{prf}} + 2 \cdot S_{\mathtt{Sym}} + n \cdot S_{\mathtt{prp}})$, which is also efficient for health monitoring systems. Thus, the PPDT achieves storage and communication efficiency.

B. Experimental Settings

We conduct experimental evaluations in the real data set to show the performance advantage of PPDT. PPDT is implemented in C++ code based on OpenSSL.[1] The experiments are conducted on a 64-bit VMware Workstation (running Ubuntu 18.04) with an Intel Core i7-8850H CPU with 2.60-GHz and 8-GB RAM. AES-CBC-256 and HMAC-256 are utilized to implement Sym and prf, respectively. The prp is implemented by utilizing prf.

Data Sets: We utilize the Breast-Cancer-Wisconsin[2] data set to train the clinical decision model for health monitoring systems. The Breast-Cancer-Wisconsin data set includes 683 nonmissing biomedical data records. Each record has nine discrete attributes, whose values are located in the integer set $\{1, \ldots, 10\}$.

Decision Trees. The decision trees are trained in plaintext-form by using scikit-learn.[3] The CART decision tree classification technique is utilized for training five decision

[1] https://www.openssl.org/

[2] http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/

[3] https://scikit-learn.org

TABLE IV
PPDT CLASSIFICATION SCHEMES AND THEIR PERFORMANCE PROPERTIES

Performance Properties	PPDT	The Scheme in [20] (SDTC)
Cost of Initialization	$T'_{\mathtt{prf}} + T_{\mathtt{gen}} + 2 \cdot T'_{\mathtt{prp}}$	$T'_{\mathtt{prf}} + T_{\mathtt{gen}} + 2 \cdot T'_{\mathtt{prp}}$
Cost of Classifier Encryption	$m \cdot n \cdot w \cdot (T_{\mathtt{prf}} + T_{\mathtt{prp}}) + m \cdot (T_{\mathtt{enc}} + T_{\mathtt{prp}})$	$(w^n) \cdot (2 \cdot T_{\mathtt{prp}} + T_{\mathtt{prf}} + T_{\mathtt{enc}} + T_{\mathtt{XOR}})$
Cost of Token Generation	$T_{\mathtt{gen}} + m \cdot ((n+1)T_{\mathtt{XOR}} + n \cdot (T_{\mathtt{prf}} + T_{\mathtt{prp}}) + 2 \cdot T_{\mathtt{enc}})$	$T_{\mathtt{prp}} + T_{\mathtt{prf}}$
Cost of Evaluation	$m \cdot (n+1) \cdot T_{\mathtt{XOR}} + (\mathcal{O}(n)+2) \cdot T_{\mathtt{dec}}$	$T_{\mathtt{XOR}} + T_{\mathtt{dec}}$
Size of Indexes	$m \cdot n \cdot w \cdot (S_{\mathtt{prf}}) + m \cdot S_{\mathtt{Sym}}$	$2 \cdot (w^n) \cdot (S_{\mathtt{Sym}} + S_{\mathtt{prf}})$
Size of Tokens	$m \cdot (S_{\mathtt{prf}} + 2 \cdot S_{\mathtt{Sym}} + n \cdot S_{\mathtt{prp}})$	$S_{\mathtt{prf}} + S_{\mathtt{prp}}$

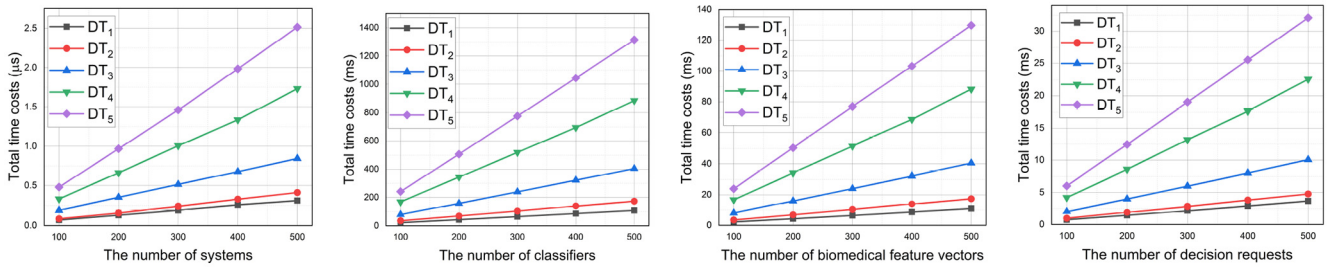


Fig. 5. Time costs of PPDT. (a) Time cost of initiation. (b) Time cost of classifier encryption. (c) Time cost of token generation. (d) Time cost of evaluation.

TABLE V
PRECISION OF THE TEST CLINICAL DECISION MODELS

Models	TP	FP	FN	TN	Accuracy
DT_1	423	10	21	229	95.46%
DT_2	423	6	21	233	96.05%
DT_3	422	3	22	236	96.34%
DT_4	433	6	11	233	97.51%
DT_5	434	5	10	234	97.80%

trees from the Breast-Cancer-Wisconsin data set. The numbers of leaf nodes (m), decision nodes (n), and plaintext domain (w) are listed as follows: 1) DT_1: $m = 4$, $n = 3$, and $w = 10$; 2) DT_2: $m = 5$, $n = 4$, and $w = 10$; 3) DT_3: $m = 7$, $n = 6$, and $w = 10$; 4) DT_4: $m = 10$, $n = 9$, and $w = 10$; and 5) DT_5: $m = 12$, $n = 11$, and $w = 10$. We provide Table V to show the true positive (TP), false positive (FP), false negative (FN), true negative (TN), and decision accuracy of the aforementioned decision trees. As shown in Table V, although parameters, such as m, n, and w are small, the accuracy of each model is high (achieves an accuracy that more than 95%). In real-world applications, if these parameters are larger, the clinical decision model may become overfitting. Thus, we utilize the above five clinical decision model to evaluate the performance advantages of PPDT.

Baselines: We compare the performance advantages of PPDT with the scheme in [20] (SDTC). Similar to PPDT, SDTC also achieves $\mathcal{O}(1)$ computational complexity and is designed based on symmetric key encryption, Prfs, and Prps. The main difference between PPDT and SDTC is the construction of indices. As shown in Table IV, the size of indices in PPDT is polynomial to m, n, and w, while the size of indices in SDTC is exponential to w. Although m, n, and w are constants, the computational, communication, and storage overheads of PPDT and SDTC are different. In the comparison experiments, SDTC is also implemented in the same setting as PPDT.

C. Experimental Evaluations

We make 500 experiment runs to evaluate the total time cost of each algorithm of PPDT, which are shown in Fig. 5. For each algorithm, we test DT_1, DT_2, DT_3, DT_4, and DT_5 that are trained from the Breast-Cancer-Wisconsin data set. Fig. 5(a) illustrates that the total time cost of the initiation algorithm grows linearly when the number of systems grows linearly, which demonstrates that the time complexity of the Init algorithm of PPDT is $\mathcal{O}(1)$. Fig. 5(b) shows that the total time cost of the classifier encryption algorithm grows linearly when the number of decision tree classifiers grows linearly, which illustrates that the time complexity of the ClfEnc algorithm of PPDT is $\mathcal{O}(1)$. Fig. 5(c) demonstrates that the total time cost of the token generation algorithm grows linearly when the number of biomedical feature vectors grows linearly, which describes that the time complexity of the TokenGen algorithm of PPDT is $\mathcal{O}(1)$. Fig. 5(d) demonstrates that the total time cost of the evaluation algorithm grows linearly when the number of decision requests grows linearly, which shows that the time complexity of the Eva algorithm of PPDT is $\mathcal{O}(1)$. Therefore, Fig. 5 shows that the time complexity of each algorithm in PPDT is $\mathcal{O}(1)$, which demonstrates that PPDT achieves the faster-than-linear time complexity.

We compare PPDT with SDTC, which enables $\mathcal{O}(1)$ computational complexity for secure decision tree classification. We evaluate performance differences between PPDT and SDTC by testing the average time costs at hospital ($\mathcal{H}$), cloud service provider ($\mathcal{CSP}$), and clients ($\mathcal{C}$), the communication costs at $\mathcal{H}$ and $\mathcal{C}$, and the storage costs at $\mathcal{CSP}$. Table VI shows the performance advantages of PPDT compared with SDTC on the Breast-Cancer-Wisconsin data set. Note that due to the heavy storage and communication overhead, we estimate the performance of SDTC in case DT_4 and DT_5 by utilizing the results in Table IV and the tested results of parameters in Table III.

Compared with SDTC, Table VI shows that PPDT boosts the efficiency in terms of average time costs. With polynomial

TABLE VI
PERFORMANCE DIFFERENCES IN THE BREAST CANCER WISCONSIN DATA SET FOR HEALTH MONITORING SYSTEMS

Models	Schemes	Time Costs				Communication Costs		Storage Costs
		$\mathcal{H}$	$\mathcal{CSP}$	$\mathcal{C}$	Total	$\mathcal{H}$	$\mathcal{C}$	$\mathcal{CSP}$
DT_1	PPDT	$215.6\mu s$	$7.2\mu s$	$21.6\mu s$	$244.4\mu s$	4.0 KB	387.1 B	4.0 KB
	SDTC [20]	$1394.9\mu s$	$0.1\mu s$	$1.5\mu s$	$1396.5\mu s$	128 KB	32.4 B	128 KB
DT_2	PPDT	$348.0\mu s$	$9.4\mu s$	$34.7\mu s$	$392.1\mu s$	6.6 KB	485.8 B	6.6 KB
	SDTC [20]	$14546.4\mu s$	$0.2\mu s$	$1.8\mu s$	$14548.4\mu s$	1280 KB	32.5 B	1280 KB
DT_3	PPDT	$811.0\mu s$	$19.9\mu s$	$80.9\mu s$	$911.8\mu s$	13.7 KB	685.8 B	13.7 KB
	SDTC [20]	$165.7ms$	$0.4\mu s$	$2.2\mu s$	$165.7ms$	128 MB	32.8 B	128 MB
DT_4	PPDT	$1773.5\mu s$	$44.8\mu s$	$176.4\mu s$	$1994.7\mu s$	29.1 KB	993.2 B	29.1 KB
	SDTC [20]	~ 1500s	$\sim 0.3\mu s$	$\sim 2\mu s$	~ 1500s	~ 128 GB	~ 33.1 B	~ 128 GB
DT_5	PPDT	$2636.0\mu s$	$63.9\mu s$	$259.5\mu s$	$2959.4\mu s$	42.6 KB	1.2 KB	42.6 KB
	SDTC [20]	$\sim 1.5 \times 10^5$s	$\sim 0.3\mu s$	$\sim 2\mu s$	$\sim 1.5 \times 10^5$s	~ 12.8 TB	~ 33.4 B	~ 12.8 TB

size indices, the computational efficiency of PPDT is orders of magnitudes faster than SDTC, which utilizes indices with exponential size, in terms of total time costs. Specifically, by reducing the size of indices, PPDT improves the computational efficiency of classifier encryption algorithm and, thus, significantly reduces the time cost at the $\mathcal{H}$ side. Hence, PPDT only requires several microseconds for $\mathcal{H}$, $\mathcal{CSP}$, and $\mathcal{C}$ to finish privacy-preserving health monitoring processes with decision tree classification. Meanwhile, Table VI demonstrates that the communication costs of PPDT is about several kilobytes, which is practical for health monitoring services. For DT_5 in Table VI, which has 11 internal nodes and 12 leaf nodes, PPDT only requires 42.6-KB communication cost at $\mathcal{H}$ side and 1.2 KB at $\mathcal{C}$ side, but SDTC requires 12.8 TB at $\mathcal{H}$ side and 33.4 Bytes at $\mathcal{C}$ side. Thus, it is clear that PPDT is communication-efficient for health monitoring services. Finally, with small size indices for encrypted decision trees, PPDT significantly reduces the storage cost at the $\mathcal{CSP}$ side, which benefits health monitoring services. Therefore, PPDT is a computational, communication, and storage efficient scheme for privacy-preserving health monitoring systems.

VIII. CONCLUSION

In this article, we have proposed PPDT, which enables privacy-preserving decision tree classification for health monitoring systems. Different from the existing schemes, PPDT transforms a decision tree classifier into the Boolean vectors and utilizes symmetric encryption to protect the data privacy. With such a design, PPDT extremely boosts the computational, communication, and storage efficiency. We have formulated a leakage function $\mathcal{L}$, provided the adaptively $\mathcal{L}$-security definition, and given a simulation-based security proof for PPDT. Performance analyses demonstrated that PPDT achieves $\mathcal{O}(1)$ computational complexity with polynomial-size indices. Experimental evaluations demonstrated that PPDT achieves microsecond-level execution time, kilobyte-level communication costs, and kilobyte-level storage costs on the Breast-Cancer-Wisconsin data set. Consequently, we have addressed both the confidentiality and efficiency challenges simultaneously, and PPDT is an efficient, practical, and real-time solution for health monitoring systems. For the future works, we will design a PPDT classification scheme against malicious adversaries.

REFERENCES

[1] J. H. Abawajy and M. M. Hassan, "Federated Internet of Things and cloud computing pervasive patient health monitoring system," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 48–53, Jan. 2017.

[2] D. B. Neill, "Using artificial intelligence to improve hospital inpatient care," *IEEE Intell. Syst.*, vol. 28, no. 2, pp. 92–95, Mar. 2013.

[3] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang, "Data mining for Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 77–97, 1st Quart., 2014.

[4] H. Habibzadeh, K. Dinesh, O. R. Shishvan, A. Boggio-Dandry, G. Sharma, and T. Soyata, "A survey of Healthcare Internet-of-Things (HIoT): A clinical perspective," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 53–71, Jan. 2020, doi: 10.1109/JIOT.2019.2946359.

[5] R. Benlamri and L. Docksteader, "MORF: A mobile health-monitoring platform," *IT Prof.*, vol. 12, no. 3, pp. 18–25, May/Jun. 2010.

[6] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "HealthDep: An efficient and secure deduplication scheme for cloud-assisted ehealth systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4101–4112, Sep. 2018.

[7] J. Hua *et al.*, "CINEMA: Efficient and privacy-preserving online medical primary diagnosis with skyline query," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1450–1461, Apr. 2019.

[8] A. Yang, J. Xu, J. Weng, J. Zhou, and D. S. Wong, "Lightweight and privacy-preserving delegatable proofs of storage with data dynamics in cloud storage," *IEEE Trans. Cloud Comput.*, vol. 9, no. 1, pp. 212–225, Jan./Mar. 2021, doi: 10.1109/TCC.2018.2851256.

[9] Y. Zhang, C. Xu, X. Lin, and X. S. Shen, "Blockchain-based public integrity verification for cloud storage against procrastinating auditors," *IEEE Trans. Cloud Comput.*, early access, Mar. 29, 2019, doi: 10.1109/TCC.2019.2908400.

[10] A. Khedr, G. Gulak, and V. Vaikuntanathan, "SHIELD: Scalable homomorphic implementation of encrypted data-classifiers," *IEEE Trans. Comput.*, vol. 65, no. 9, pp. 2848–2858, Sep. 2016.

[11] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proc. NDSS*, 2015, pp. 1–34.

[12] D. J. Wu, T. Feng, M. Naehrig, and K. E. Lauter, "Privately evaluating decision trees and random forests," in *Proc. Privacy Enhanc. Technol.*, vol. 2016, no. 4, pp. 335–355, 2016.

[13] R. K. H. Tai, J. P. K. Ma, Y. Zhao, and S. S. M. Chow, "Privacy-preserving decision trees evaluation via linear functions," in *Proc. ESORICS*, 2017, pp. 494–512.

[14] L. Xue, D. Liu, C. Huang, X. Lin, and X. S. Shen, "Secure and privacy-preserving decision tree classification with lower complexity," *J. Commun. Inf. Netw.*, vol. 5, no. 1, pp. 16–25, Mar. 2020.

[15] K. A. Jagadeesh, D. J. Wu, J. A. Birgmeier, D. Boneh, and G. Bejerano, "Deriving genomic diagnoses without revealing patient genomes," *Science*, vol. 357, no. 6352, pp. 692–695, 2017.

[16] M. D. Cock *et al.*, "Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 2, pp. 217–230, Mar./Apr. 2019.

[17] A. Tueno, F. Kerschbaum, and S. Katzenbeisser, "Private evaluation of decision trees using sublinear cost," *Proc. Privacy Enhanc. Technol.*, vol. 2019, no. 1, pp. 266–286, 2019.
[18] Á. Kiss, M. Naderpour, J. Liu, N. Asokan, and T. Schneider, "SoK: Modular and efficient private decision tree evaluation," *Proc. Privacy Enhanc. Technol.*, vol. 2019, no. 2, pp. 187–208, 2019.
[19] Y. Zheng, H. Duan, and C. Wang, "Towards secure and efficient outsourcing of machine learning classification," in *Proc. ESORICS*, 2019, pp. 22–40.
[20] J. Liang, Z. Qin, S. Xiao, L. Ou, and X. Lin, "Efficient and secure decision tree classification for cloud-assisted online diagnosis services," *IEEE Trans. Dependable Secure Comput.*, early access, Jun. 14, 2019, doi: 10.1109/TDSC.2019.2922958.
[21] W. W. Cohen, "Fast effective rule induction," in *Proc. ICML*, 1995, pp. 115–123.
[22] N. Cheng *et al.*, "Big data driven vehicular networks," *IEEE Netw.*, vol. 32, no. 6, pp. 160–167, Nov./Dec. 2018.
[23] H. Yang, Q. Zhou, M. Yao, R. Lu, H. Li, and X. Zhang, "A practical and compatible cryptographic solution to ADS-B security," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3322–3334, Apr. 2019.
[24] F. Lyu *et al.*, "Characterizing urban vehicle-to-vehicle communications for reliable safety applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2586–2602, Jun. 2020, doi: 10.1109/TITS.2019.2920813.
[25] W.-D. Yang and T. Wang, "The fusion model of intelligent transportation systems based on the urban traffic ontology," *Phys. Procedia*, vol. 25, pp. 917–923, 2012.
[26] C. Huang, R. Lu, X. Lin, and X. Shen, "Secure automated valet parking: A privacy-preserving reservation scheme for autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11169–11180, Nov. 2018.
[27] G. Xiao, J. Li, Y. Chen, and K. Li, "MalFCS: An effective malware classification framework with automated feature extraction based on deep convolutional neural networks," *J. Parallel Distrib. Comput.*, vol. 141, pp. 49–58, Jul. 2020.
[28] W. Tang, J. Ren, K. Deng, and Y. Zhang, "Secure data aggregation of lightweight e-healthcare IoT devices with fair incentives," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8714–8726, Oct. 2019.
[29] J. Liang, Z. Qin, S. Xiao, J. Zhang, H. Yin, and K. Li, "Privacy-preserving range query over multi-source electronic health records in public clouds," *J. Parallel Distrib. Comput.*, vol. 135, pp. 127–139, Jan. 2020.
[30] C. Zhang, L. Zhu, C. Xu, and R. Lu, "PPDP: An efficient and privacy-preserving disease prediction scheme in cloud-based e-healthcare system," *Future Gener. Comput. Syst.*, vol. 79, pp. 16–25, Feb. 2018.
[31] W. Tang, K. Zhang, J. Ren, Y. Zhang, and X. Shen, "Flexible and efficient authenticated key agreement scheme for bans based on physiological features," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 845–856, Apr. 2019.
[32] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in Internet of Things with privacy preserving: Challenges, solutions and opportunities," *IEEE Netw.*, vol. 32, no. 6, pp. 144–151, Nov./Dec. 2018.
[33] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Boca Raton, FL, USA: Chapman & Hall/CRC, 2007.
[34] S. Lai *et al.*, "Result pattern hiding searchable encryption for conjunctive queries," in *Proc. ACM SIGSAC*, 2018, pp. 745–762.
[35] J. Liang, Z. Qin, J. Ni, X. Lin, and X. S. Shen, "Practical and secure SVM classification for cloud-based remote clinical decision services," *IEEE Trans. Comput.*, early access, Sep. 1, 2020, doi: 10.1109/TC.2020.3020545.
[36] S. Wu, Q. Li, G. Li, D. Yuan, X. Yuan, and C. Wang, "ServeDB: Secure, verifiable, and efficient range queries on outsourced database," in *Proc. IEEE ICDE*, 2019, pp. 626–637.

Jinwen Liang (Graduate Student Member, IEEE) received the B.S. degree in information security from Hunan University, Changsha, China, in 2015, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering.

He is also a visiting Ph.D. student with BBCR Laboratory, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include applied cryptography, blockchain, order-preserving encryption, and secure data classification.

Mr. Liang served as the TPC Member for IEEE VTC 2019 Fall.

Zheng Qin (Member, IEEE) received the Ph.D. degree in computer science from Chongqing University, Chongqing, China, in 2001.

He was a Visiting Scholar with Michigan State University, East Lansing, MI, USA, from 2010 to 2011. He is a Full Professor and the Vice Dean with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. He is the Director of the Hunan Key Laboratory of Big Data Research and Application, and the Vice Director of the Hunan Engineering Laboratory of Authentication and Data Security, Changsha. His research interests include blockchain, data science, information security, and software engineering.

Liang Xue received the B.S. and M.S. degrees from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2015 and 2018, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

Her research interests include applied cryptography, cloud computing, and blockchain.

Xiaodong Lin (Fellow, IEEE) received the Ph.D. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1998, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2008.

He is currently an tenured Associate Professor with the School of Computer Science, University of Guelph, Guelph, ON, Canada. His research interests include wireless network security, applied cryptography, computer forensics, and software security.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular *ad hoc* and sensor networks.

Dr. Shen received the R. A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and the Education Award in 2017 from the IEEE Communications Society, and the Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award from the University of Waterloo in 2006, and the Premier's Research Excellence Award from the Province of Ontario, Canada, in 2003. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, and IEEE Globecom'07, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is the President Elect of the IEEE Communications Society. He was the Vice President for Technical and Educational Activities and for Publications, a Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a Member of IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*. He is a registered Professional Engineer of Ontario, Canada, a Fellow of Engineering Institute of Canada, Canadian Academy of Engineering, Royal Society of Canada, a Foreign Member of Chinese Academy of Engineering, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.