Verifiable and Secure SVM Classification for Cloud-Based Health Monitoring Services

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

***Abstract*—In cloud-based health monitoring services, support vector machine (SVM) classification techniques are often utilized by medical institutes to build medical decision models, which can be outsourced to a cloud server for producing medical decisions based on medical features from remote clients. In this article, we propose a verifiable and secure SVM classification scheme (VSSVMC) for cloud-based health monitoring services in a malicious setting, where the cloud server may return invalid decisions. By constructing verifiable indices, VSSVMC ensures the verifiability of medical decisions, which enables clients to detect whether the cloud server returns incorrect or incomplete medical decisions. Symmetric key encryption is leveraged to ensure the confidentiality of the medical decision model and medical data with computational efficiency. We give security and verifiability definitions and provide formal security and verifiability proofs for VSSVMC. Performance analyses show that VSSVMC is extremely efficient in terms of computation, communication, and storage. Experimental evaluations demonstrate that VSSVMC achieves microsecond-level execution time with kilobyte-level communication and storage overheads on the tested data set.**

***Index Terms*—Cloud-based health monitoring services, secure support vector machine (SVM) classification, verification.**

I. Introduction

IN THE past decades, the growth of chronic diseases and the elderly have significantly raised the costs of medical services [1]. With the proliferation of artificial intelligence and wearable devices, numerous medical institutes provide health monitoring services to reduce skyrocketing healthcare costs and improve the quality of medical decision services [2]. Specifically, health monitoring services, which are often built by utilizing support vector machine (SVM) classification techniques, enable medical institutes to monitor clients' symptoms periodically and produce real-time medical decisions based on the collected features and the pretrained SVM decision model [3]–[5].

Coupled with the recent advances of cloud computing, cloud-based health monitoring services are new options to further reduce computational and storage costs in health monitoring systems [6], [7]. In particular, cloud-based health monitoring services involve three entities, i.e., a medical institute, a cloud server, and clients [8]. Specifically, the medical institute outsources a medical decision model to a cloud server and, later, clients submit their medical features to the cloud server periodically and receive their real-time medical decisions based on the outsourced model. Such a procedure brings many benefits for health monitoring services, such as ease of management, ubiquitous access, and scalability [9].

Apart from the well-known advantages, cloud-based health monitoring services may lead to critical privacy concerns as the cloud server may not behave honestly [10], [11]. From the perspective of medical institutes, the medical decision model, which is trained from a significant amount of sensitive medical records, is a valuable knowledge asset. Due to intellectual property protection issues, the confidentiality of the medical decision model should be ensured [12]. From the perspective of clients, both medical features and medical decisions are sensitive medical data for them. Accidental exposure of medical data may increase health insurance costs when a client has chronic diseases. Furthermore, due to reasons, such as software errors, internal attacks, external attacks, and monetary issues, the cloud server could be compromised and then behave maliciously [13], [14]. For example, the cloud server may return incorrect or incomplete medical decisions for saving the computational resources or reducing the communication costs. Such malicious behaviors may lead to misdiagnosis due to false medical decisions. Hence, the confidentiality of

Manuscript received January 20, 2021; revised March 30, 2021; accepted April 17, 2021. Date of publication April 26, 2021; date of current version November 19, 2021. This work was supported in part by the China Scholarship Council under Grant 201806130132; in part by the National Natural Science Foundation of China under Grant 61772191, Grant U20A20174, Grant 62002112, and Grant 61902123; in part by the Science and Technology Key Projects of Hunan Province under Grant 2018TP2023, Grant 2019WK2072, and Grant 2015TP1004; in part by the National Key Research and Development Projects under Grant 2018YFB0704000; in part by the China Postdoctoral Science Foundation under Grant 2020M672488; in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada; in part by the Science and Technology Key Projects of Changsha City under Grant kq2004027, Grant kq2004025, and Grant kq2006029; and in part by the Natural Science Foundation of Hunan Province under Grant 2020JJ5085. *(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.2021.3075540

both medical decision models and medical data should be guaranteed, and the verifiability of medical decisions should be enabled.

In order to ensure the confidentiality, many secure SVM classification schemes have been proposed [15]–[23]. The existing schemes are mainly designed based on homomorphic encryption (HE) [15]–[17], bilinear pairing [18], secure multiparty computation (MPC) [19], [20], order-preserving encryption (OPE) [21], matrix transformation (MT) [22], and randomized Bloom filters [23]. Some of the aforementioned schemes may incur prohibitive computational costs [15]–[18], lead to high communication costs [19], [20], reveal numerical orders of medical data [21], leak the distribution of medical data [22], and introduce false-positive rates to medical decisions [23]. Furthermore, the aforementioned schemes assume that the cloud server is a honest-but-curious adversary [21], [23]. Unfortunately, this assumption does not always hold in real-world applications, because the cloud server may deviate from the prescribed scheme [24]. Therefore, it is desirable to achieve the verifiability of the returned decisions in secure SVM classification for constructing trustworthy and privacy-preserving health monitoring services.

In this article, we proposed a verifiable and secure SVM classification scheme (VSSVMC) for cloud-based health monitoring services in a *malicious* setting, which is a stronger threat model than that of previous secure SVM classification schemes. Different from the popular adopted honest-but-curious threat model, the malicious threat model enables a cloud server to forge or delete some of the medical decisions. To design VSSVMC, we first transform the SVM classification functionality to a function of conjunctively querying whether a feature vector is located in a multidimensional interval. Then, we build efficient query indices for the SVM classifier, which could be expressed by decision rules. Later, we leverage pseudorandom permutations, pseudorandom functions, and symmetric key encryption to encrypt the query indices and produce pseudorandom strings for decision verification. After that, the VSSVMC could be achieved by querying such encrypted indices. By ensuring both decision verifiability and data confidentiality, VSSVMC enables trustworthy and secure cloud-based health monitoring services. The contributions of this article are shown as follows.

1) We consider a malicious threat model and propose VSSVMC for cloud-based health monitoring services. By leveraging pseudorandom permutations, pseudorandom functions, and symmetric key encryption, VSSVMC ensures the verifiability of medical decisions and the confidentiality of both the medical decision model and the medical data. Thus, VSSVMC is secure against a malicious adversary, which may forge or delete the medical decisions in cloud-based health monitoring services. Furthermore, VSSVMC is computation, communication, and storage efficient due to the construction of efficient query indices and the adoption of lightweight cryptographic primitives.
2) We give security and verifiability definitions and provide formal proofs for VSSVMC. First, we define a leakage function $\mathcal{L}$ to evaluate the information leakage of VSSVMC, which contains size patterns, access patterns, and search patterns. Second, we give the $\mathcal{L}$-security definition and provide a simulation-based security proof to show VSSVMC is $\mathcal{L}$-secure. Third, we provide the correctness and completeness definitions and define the verifiability for VSSVMC. Finally, we provide a game-based verifiability proof to demonstrate that VSSVMC ensures the decision verifiability.
3) Compared with existing secure SVM classification schemes, VSSVMC ensures verifiability and confidentiality with competitive performance in terms of computation, communication, and storage costs. The performance analyses show the computational, communication, and storage costs of VSSVMC are similar to that of the nonverifiable scheme in [23] (*LQNLS20*). The experimental evaluations are conducted on the Breast-Cancer-Wisconsin data set. The performance evaluation results show that VSSVMC: a) achieves $\mathcal{O}(1)$ computational complexity when an SVM classifier is pretrained; b) achieves microsecond execution time for each algorithm; and c) requires tiny communication costs (less than 6 kB) and storage costs (less than 47 kB).

The remainder of this article is organized as follows. Section II presents the related work. Section III describes the system model, threat model, and design goals. Section IV provides the preliminaries. Section V illustrates the detail construction of VSSVMC. Section VI gives the security and verifiability definitions and provides formal security and verifiability proofs. Section VII analyzes and evaluates the performance of VSSVMC. Section VIII concludes this article.

II. Related Work

With the advantages, such as efficient evaluation, ease of deployment, and high accuracy, data classification techniques have been widely deployed for making decisions in many application fields, such as healthcare [25], transportation [26]–[29], and so on. In cloud-based health monitoring services, SVM classification is often utilized to construct a medical decision model, which later is outsourced to a remote cloud server for producing medical decisions based on medical features collected from clients. Since the cloud server may be compromised due to internal or external attacks, the confidentiality of medical decision models and medical data should be protected.

To ensure the confidentiality of both medical models and medical data in cloud-based health monitoring services, a significant amount of secure SVM classification schemes have been proposed. The existing schemes can be categorized as HE-based schemes [15]–[17], bilinear pairing-based schemes [18], secure MPC-based schemes [19], [20], OPE-based schemes [21], MT-based schemes [22], and randomized Bloom filters-based schemes [23]. Due to time-consuming operations in HE, MPC, and pairing, some of these schemes may incur prohibitive computational costs [15]–[18] or communication costs [19], [20] to resource-limited body sensors and wearable devices used in cloud-based health monitoring services. Both OPE-based [21] and MT-based [22] schemes

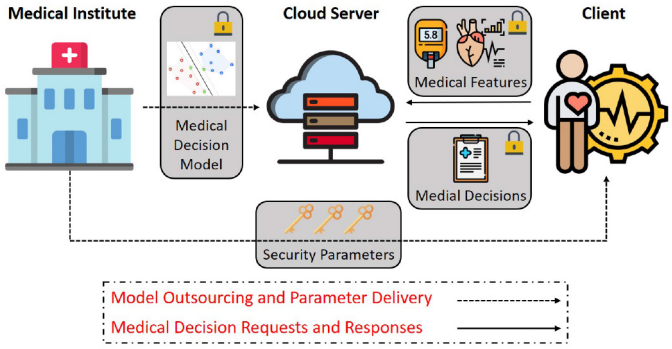

Fig. 1. System model of cloud-based health monitoring services.

are lightweight in terms of computational, communication, and storage overheads. Yet, OPE-based schemes may leak the numerical orders of data [21] and MT-based schemes may reveal the distribution of data [22]. Thus, both OPE-based schemes [21] and MT-based [22] schemes may incur privacy leakage in cloud-based health monitoring services. The randomized Bloom filter-based scheme in [23] protects the confidentiality of medical data with computational and communication efficiency. However, Bloom filter techniques inevitably introduce false positive to the medical decisions, which may lead to misdiagnosis issues.

More importantly, the aforementioned schemes are designed based on the assumptions that the adversaries in health monitoring services are honest-but-curious, which means they execute the protocol faithfully but are curious about the content of both medical models and medical data [23]. Unfortunately, this assumption may become invalid in practice due to internal and external attacks [30], software errors [31], etc. All these issues may lead the cloud server to forge or delete some of the medical decisions [32], [33]. However, none of the aforementioned schemes enables verifiability of medical decision to ensure the cloud server faithfully provides the health monitoring services. Therefore, it is desirable to design an efficient, verifiable, and secure SVM classification scheme for providing lightweight, trustworthy, and privacy-preserving health monitoring services.

III. Models and Design Goals

A. System Model

We provide the system model of cloud-based health monitoring services in Fig. 1, which contains three entities, i.e., a medical institute ($\mathcal{MI}$), a cloud server ($\mathcal{CS}$), and a client ($\mathcal{C}$). Each entity is described as follows.

1) *Medical Institute ($\mathcal{MI}$):* $\mathcal{MI}$ owns a medical decision model, which is pretrained via SVM classification techniques, for cloud-based health monitoring services. To enjoy the benefits of cloud computing, $\mathcal{MI}$ will stay offline after outsourcing the medical decision model to $\mathcal{CS}$.
2) *Cloud Server ($\mathcal{CS}$):* $\mathcal{CS}$ is a third-party service provider, which enables efficient and flexible cloud-based services. With the medical decision model (the pretrained SVM classifier) from $\mathcal{MI}$, $\mathcal{CS}$ provides health monitoring services for $\mathcal{C}$ by making medical decisions based on $\mathcal{C}$'s medical features.
3) *Client ($\mathcal{C}$):* $\mathcal{C}$ is a client who needs to test his personal health condition periodically. Due to computation and network resource constraints, he will stay offline after submitting his medical features to $\mathcal{CS}$ and later retrieve the medical decisions.

With the above three entities, the procedure of cloud-based health monitoring services contains two stages.

1) *Model Outsourcing and Parameter Delivery:* In this stage, $\mathcal{MI}$ interacts with $\mathcal{CS}$ and $\mathcal{C}$ only once. Namely, $\mathcal{MI}$ outsources the medical decision model to $\mathcal{CS}$ and delivers some parameters to $\mathcal{C}$.
2) *Medical Decision Requests and Responses:* In this stage, $\mathcal{C}$ interacts with $\mathcal{CS}$ periodically. Namely, $\mathcal{C}$ submits multiple medical decision requests to $\mathcal{MI}$ and waiting for the corresponding medical decisions, for each request-response round: $\mathcal{C}$ submits his medical features to $\mathcal{CS}$ and $\mathcal{CS}$ returns the corresponding medical decisions by utilizing the medical decision model.

B. Threat Model

In cloud-based health monitoring services, both $\mathcal{MI}$ and $\mathcal{C}$, which have medical decision model and medical data (including both medical features and medical decisions), respectively, are always viewed as *honest* entities. As a result, both $\mathcal{MI}$ and $\mathcal{C}$ will not collude with $\mathcal{CS}$. Since $\mathcal{CS}$ may be fully compromised or behave maliciously, we consider $\mathcal{CS}$ as a *malicious* adversary, which can conduct the following behaviors.

1) $\mathcal{CS}$ may try to derive some sensitive information when providing health monitoring services. Namely, $\mathcal{CS}$ may try to steal the medical decision model or record the medical data for monetary reasons.
2) $\mathcal{CS}$ may return invalid (incomplete or incorrect) medical decisions to $\mathcal{C}$. Namely, $\mathcal{CS}$ may forge or delete some of the medical decision results for saving the computational resources or reducing the communication costs.

C. Design Goals

We aim to propose VSSVMC, which ensures the confidentiality, verifiability, and efficiency for cloud-based health monitoring services. We consider the following properties for VSSVMC.

1) *Confidentiality:* Both the medical decision model and the medical data (including medical features and medical decisions) should be protected against $\mathcal{CS}$. With confidentiality, both the intellectual property (medical decision model) of $\mathcal{MI}$ and the data privacy (both medical features and decisions) of $\mathcal{C}$ are protected in cloud-based health monitoring services, which alleviates both the intellectual property protection and privacy leakage concerns of $\mathcal{MI}$ and $\mathcal{C}$, respectively.
2) *Verifiability:* The invalid (including incorrect or incomplete) medical decisions received from $\mathcal{CS}$ should be detected. With verifiability, the cloud-based health monitoring service is secure against malicious adversaries and robust against software errors, which ensures the accuracy of medical decisions for $\mathcal{C}$.

3) *Efficiency:* Sublinear computational complexity and microsecond-level execution time should be achieved. With efficiency, the secure cloud-based health monitoring service is lightweight and efficient for resource-limited body sensors and wearable devices.

IV. Preliminaries

A. Cryptographic Preliminaries

We utilize pseudorandom functions (prf), pseudorandom permutations (prp), and symmetric key encryption (Sym) to construct VSSVMC. Pseudorandom functions are keyed functions, whose outputs are computationally indistinguishable from randomly selected strings. More details of prf could be found in [8], [23], and [34]. Pseudorandom permutations denotes keyed bijections, whose output is a permutation which cannot not be computationally distinguished from a randomly selected permutation from the set of all permutations in the functions' domain. More details of prp could be found in [8], [23], and [34]. Symmetric key encryption denotes any encryption, whose encryption key is the same as the decryption key. We assume that Sym is IND-CPA secure [34] in this article. More details of Sym could be found in [8], [23], and [34].

B. Support Vector Machine and Rule Extraction

SVM is a binary data classification technique with high precision, which has been widely utilized in health monitoring services [16], [18], [23]. Let $\boldsymbol{v} = \{v_1, v_2, \ldots, v_m\}$ be an m-dimensional input feature. We assume that each $v_i \in \boldsymbol{v}$ is normalized to an interval $[0, n]$ by utilizing normalization techniques. Let $p = \{-1, +1\}$ be two different predictions, i.e., negative and positive. An SVM classifier is a separating hyperplane with a maximum margin in the m-dimensional feature space, which divides the m-dimensional feature space into two subspaces, i.e., a subspace for positive prediction and the other for negative prediction. The SVM classifier (i.e., the separating hyperplane) makes prediction by judging which subspace the input m-dimensional feature belongs to [23], [35]. More details about how the SVM classification technique works can be found in [16], [18], and [23]. Fu *et al.* [35] developed an SVM rule extraction scheme by producing several hyperrectangles that cover each subspace produced by the SVM classifier. Then, the boundary of hyperrectangles could be viewed as decision rules for an SVM classifier.

We provide Fig. 2 as an example to show the main idea of SVM classification and Fu *et al.*'s rule extraction. In Fig. 2, there are two dimensions for the input feature (i.e., $\boldsymbol{v} = \{v_1, v_2\}$), where the values are normalized to the interval $[0, n]$. Two predictions, i.e., positive (the blue one) and negative (the red one), are also shown in Fig. 2. After the training phase, an SVM classifier (i.e., the black solid curve in Fig. 2) is produced, which divides the 2-D feature space into two subspaces. With Fu *et al.*'s rule extraction scheme, four hyperrectangles (i.e., R_1, R_2, R_3, and R_4) are produced to cover the subspace for positive prediction, i.e., the subspace that is under the black solid curve. Meanwhile, $R_5, \ldots, R_9$ are produced to cover the subspace for negative prediction, i.e., the subspace that is above the black solid curve. For each

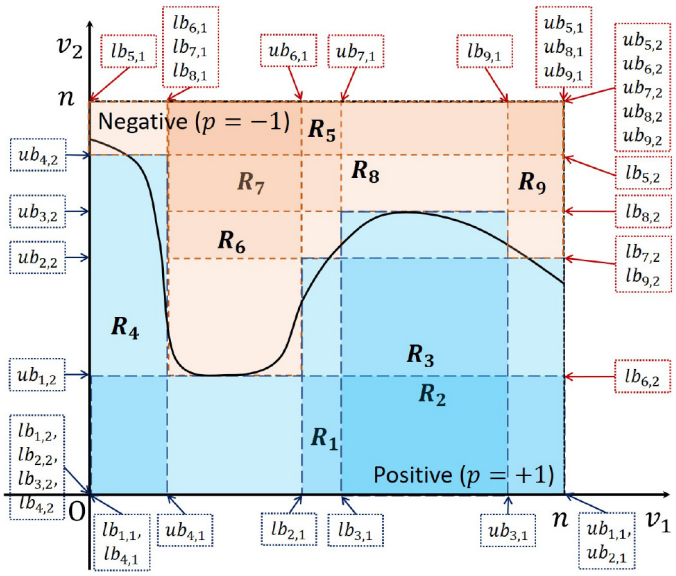

Fig. 2. Example of an SVM classifier and extracted rules.

TABLE I
Extracted Rules for Positive Prediction in Fig. 2

Rules	Conditions	Predictions
R_1	$lb_{1,1} \le v_1 \le ub_{1,1}$ and $lb_{1,2} \le v_2 \le ub_{1,2}$	Positive
R_2	$lb_{2,1} \le v_1 \le ub_{2,1}$ and $lb_{2,2} \le v_2 \le ub_{2,2}$	Positive
R_3	$lb_{3,1} \le v_1 \le ub_{3,1}$ and $lb_{3,2} \le v_2 \le ub_{3,2}$	Positive
R_4	$lb_{4,1} \le v_1 \le ub_{4,1}$ and $lb_{4,2} \le v_2 \le ub_{4,2}$	Positive
R_5	$lb_{5,1} \le v_1 \le ub_{5,1}$ and $lb_{5,2} \le v_2 \le ub_{5,2}$	Negative
R_6	$lb_{6,1} \le v_1 \le ub_{6,1}$ and $lb_{6,2} \le v_2 \le ub_{6,2}$	Negative
R_7	$lb_{7,1} \le v_1 \le ub_{7,1}$ and $lb_{7,2} \le v_2 \le ub_{7,2}$	Negative
R_8	$lb_{8,1} \le v_1 \le ub_{8,1}$ and $lb_{8,2} \le v_2 \le ub_{8,2}$	Negative
R_9	$lb_{9,1} \le v_1 \le ub_{9,1}$ and $lb_{9,2} \le v_2 \le ub_{9,2}$	Negative

dimension i ($i \in [1, 2]$) of each hyperrectangle R_t ($t \in [1, 9]$), the lower boundary and the upper boundary are $lb_{t,i}$ and $ub_{t,i}$, respectively. For example, the lower boundary and the upper boundary of R_2 in dimension 1 are $lb_{2,1}$ and $ub_{2,1}$, respectively. With $R_1, R_2, \ldots, R_9$, nine decision rules (also known as, the boundary of hyperrectangles) can be extracted from the SVM classifier. We show the extracted rules in Table I. To determine whether the prediction of an input feature $\boldsymbol{v}$ is positive, we only need to judge whether $\boldsymbol{v}$ satisfies the rules in Table I. For example, if $\boldsymbol{v} = \{v_1, v_2\}$ satisfies both $lb_{4,1} \le v_1 \le ub_{4,1}$ and $lb_{4,2} \le v_2 \le ub_{4,2}$, then $\boldsymbol{v}$ satisfies R_4 and the corresponding prediction is positive. Note that the boundaries of hyperrectangles may overlap in some dimensions. For example, the boundary of hyperrectangle R_2 overlaps the boundary of R_1 and R_3 in dimension v_1. Thus, when a value locates in several hyperrectangles simultaneously, all the corresponding predictions will be returned. For example, when the m-dimensional input feature $\boldsymbol{v}$ matches both R_2 and R_3 in Fig. 2, then the corresponding predictions of R_2 and R_3 are returned, i.e., positive and positive, though these two predictions are the same. This issue could be addressed by selecting the most voted predictions, which is out of the scope of this article.

TABLE II
NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
SVM	The SVM classifier trained from biomedical data.
t	the number of extracted rules.
m	the number of dimensions of input biomedical feature.
$\boldsymbol{v}$	$\boldsymbol{v} = \{v_1, \ldots, v_m\}$ is an m-dimensional feature.
n	Each $v_i \in \boldsymbol{v}$ are normalized to an interval $[0, n]$.
$\boldsymbol{R}$	$\boldsymbol{R} = \{R_1, \ldots, R_t\}$ are t rules extracted from SVM.
$\boldsymbol{p}$	$\boldsymbol{p} = \{p_1, \ldots, p_t\}$ are t corresponding predictions for $\boldsymbol{R}$.
$\boldsymbol{p}(\boldsymbol{v})$	$\boldsymbol{p}(\boldsymbol{v})$ is the matched decisions for $\boldsymbol{v}$.
R_i	$R_i = \{R_{i,1}, \ldots, R_{i,j}, \ldots, R_{i,m}, p_i\}$ is the i-th rule, where $R_{i,j} = \{lb_{i,j}, ub_{i,j}\}$ denotes the lower boundary and the upper boundary of R_i in dimension j, respectively.
$\boldsymbol{I}$	$\boldsymbol{I} = \{\boldsymbol{I}_{1,1}, \ldots, \boldsymbol{I}_{i,j}, \ldots, \boldsymbol{I}_{t,m}\}$ are $t \times m$ boolean vectors with n elements.
κ	The security parameter.
$F_i(x)$	The κ bit output of $\mathtt{prf}$ F with input x and κ bit key K_{f_i}, where $i = \{0, 1\}$.
$\boldsymbol{c}$	$\boldsymbol{c} = \{c_1, \ldots, c_t\}$ are $\mathtt{Sym}$ ciphertexts for $\boldsymbol{p}$.
$\boldsymbol{c}(\boldsymbol{v})$	The corresponding ciphertexts of $\boldsymbol{p}(\boldsymbol{v})$.
$\boldsymbol{vc}$	$\boldsymbol{vc} = \{vc_1, \ldots, vc_t\}$ is the verification messages of $\boldsymbol{c}$, where vc_i is the verification message of c_i.
$H_0(x)$	The $\log(tmn)$ bit output of $\mathtt{prp}$ H_0 with $\log(tmn)$ bit input x and κ bit key K_{h_0}.
$H_1(x)$	The $\log t$ bit output of $\mathtt{prp}$ H_1 with $\log t$ bit input x and κ bit key K_{h_1}.
$\mathtt{Sym}$	$\mathtt{Sym} = (\mathtt{Sym.Gen}, \mathtt{Sym.Enc}, \mathtt{Sym.Dec})$ is an IND-CPA secure symmetric key encryption, whose key-space, plaintext-space, and ciphertext-space are $\{0,1\}^\kappa$.
K_i	The key of $\mathtt{Sym}$, where $i = \{0, 1, \ldots, t\}$.
T_0	An encrypted index with tmn elements.
T_1	An encrypted index with t elements.
$\boldsymbol{TK}(\boldsymbol{v})$	$\boldsymbol{TK}(\boldsymbol{v}) = \{TK_1(\boldsymbol{v}), \ldots, TK_t(\boldsymbol{v})\}$ are t tokens for $\boldsymbol{v}$.
$TK_i(\boldsymbol{v})$	$TK_i(\boldsymbol{v}) = (\alpha_i, \beta_i, \gamma_i, \boldsymbol{L}_i)$, where $\boldsymbol{L}_i$ is m locations in T_0.
$\boldsymbol{PF}$	$\boldsymbol{PF} = \{\mathtt{PF}_1, \mathtt{PF}_2, \ldots, \mathtt{PF}_t\}$ are t proofs.

In summary, by utilizing Fu *et al.*'s scheme [35], an SVM classifier could be viewed as a set of rules, which are the boundaries of extracted hyperrectangles. Note that the boundaries of hyperrectangles are a set of ranges, i.e., the interval $[lb_{i,j}, ub_{i,j}]$, which may overlap in some dimensions. We utilize the extracted ranges to denote an SVM classifier in this article.

V. DESIGN OF VSSVMC

A. Definitions

Let $\boldsymbol{v} = \{v_1, \ldots, v_m\}$ be an m-dimensional biomedical feature, whose value are normalized to an interval $[0, n]$. We assume that all values in $\boldsymbol{v}$ are positive integers, i.e., $\boldsymbol{v} \in \mathbb{Z}_n^m$. Let SVM be an SVM classifier that is trained from a set of m-dimensional biomedical features. Let $\boldsymbol{R} = \{R_1, R_2, \ldots, R_t\}$ be t rules extracted from SVM. Let $\boldsymbol{p} = \{p_1, p_2, \ldots, p_t\}$ denote the corresponding medical predictions of $\boldsymbol{R}$. Each rule $\{R_i | 1 \le i \le t\} = \{R_{i,1}, R_{i,2}, \ldots, R_{i,j}, \ldots, R_{i,m}, p_i\}$, where $R_{i,j} = \{lb_{i,j}, ub_{i,j}\}$ denotes the lower boundary and the upper boundary of hyperrectangle R_i in dimension j, respectively. Since the extracted hyperrectangles may overlap with each other, the input feature $\boldsymbol{v}$ may match multiple predictions in $\boldsymbol{p}$. Let $\boldsymbol{p}(\boldsymbol{v})$ be the matched predictions for $\boldsymbol{v}$, i.e., $\boldsymbol{p}(\boldsymbol{v}) = \{p_i | p_i$ is the matched predictions of $\boldsymbol{v}\}$. To handle the multiple rules matched situation, the majority of predictions will be selected as the final prediction of the input feature. For example, when two positive rules and one negative rule are matched, the prediction of the input feature is positive. Let $\boldsymbol{I} = \{\boldsymbol{I}_{1,1}, \ldots, \boldsymbol{I}_{1,m}, \ldots, \boldsymbol{I}_{t,1}, \ldots, \boldsymbol{I}_{t,m}\}$ be $t \times m$ Boolean vectors, which denote the SVM decision rules extracted from SVM. Each vector $\boldsymbol{I}_{i,j} \in \boldsymbol{I}$ contains n Boolean elements, i.e., $I_{i,j}[k] = \{0, 1\}$, where $I_{i,j}[k] \in \boldsymbol{I}_{i,j}$ and $1 \le k \le n$.

Let κ be the security parameter. Let $F : \{0,1\}^* \times \{0,1\}^\kappa \to \{0,1\}^\kappa$ be a $\mathtt{prf}$, where $K_{f_0}, K_{f_1} \leftarrow \{0,1\}^\kappa$ are different keys of F. We write $F_i(x)$ instead of $F(K_{f_i}, x)$, where $i = \{0, 1\}$. Let $H_0 : \{0,1\}^\kappa \times \{0,1\}^{\log(tmn)} \to \{0,1\}^{\log(tmn)}$ and $H_1 : \{0,1\}^\kappa \times \{0,1\}^{\log t} \to \{0,1\}^{\log t}$ be two $\mathtt{prp}$, where $K_{h_0} \leftarrow \{0,1\}^\kappa$ and $K_{h_1} \leftarrow \{0,1\}^\kappa$ are keys of H_0 and H_1, respectively. We write $H_i(x)$ instead of $H(K_{h_i}, x)$, where $i = \{0, 1\}$. $\mathtt{Sym} = (\mathtt{Sym.Gen}, \mathtt{Sym.Enc}, \mathtt{Sym.Dec})$ is an IND-CPA secure symmetric key encryption, whose key-space, plaintext-space, and ciphertext-space are $\{0,1\}^\kappa$. Let $K_0, K_1, \ldots, K_t \leftarrow \mathtt{Sym.Gen}(1^\kappa)$ be $(t+1)$ keys for $\mathtt{Sym}$. Let $\boldsymbol{c} = \{c_1, c_2, \ldots, c_t\}$ be the corresponding $\mathtt{Sym}$ ciphertexts of $\boldsymbol{p}$. Let $\boldsymbol{c}(\boldsymbol{v})$ be the corresponding ciphertexts of $\boldsymbol{p}(\boldsymbol{v})$. Let $\boldsymbol{vc} = \{vc_1, vc_2, \ldots, vc_t\}$ be the verification message of $\boldsymbol{c}$, i.e., vc_i is the verification message of c_i. Let T_0 and T_1 be encrypted indices with tmn elements and t elements, respectively. $\boldsymbol{TK}(\boldsymbol{v}) = \{TK_1(\boldsymbol{v}), \ldots, TK_t(\boldsymbol{v})\}$ are t encrypted tokens for achieving the encrypted medical predictions for $\boldsymbol{v}$. Each encrypted token $TK_i(\boldsymbol{v}) = (\alpha_i, \beta_i, \gamma_i, \boldsymbol{L}_i)$, where $\boldsymbol{L}_i$ contains m locations in T_0. $\boldsymbol{PF} = \{\mathtt{PF}_1, \mathtt{PF}_2, \ldots, \mathtt{PF}_t\}$ are t proofs for encrypted medical decisions produced by $\boldsymbol{TK}(\boldsymbol{v})$. Table II summarized all notations and descriptions.

B. Indices and SVM Classifiers

To store the extracted rules $\boldsymbol{R}$, we build an index $\boldsymbol{I}$, which contains $t \times m$ Boolean vectors. Each vector $\boldsymbol{I}_{i,j}$, which has n elements, denotes the interval $[lb_{i,j}, ub_{i,j}]$ of $R_{i,j}$. The value of each element $I_{i,j}[k]$ in $\boldsymbol{I}_{i,j}$ is set as in

$$I_{i,j}[k] \leftarrow \begin{cases} 1, & \text{if } lb_{i,j} \le k \le ub_{i,j} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

With the input medical feature $\boldsymbol{v} = \{v_1, \ldots, v_j, \ldots, v_m\}$, the SVM classification proceeds as follows. For each rule R_i, if

$$\bigwedge_{j=1}^{m} I_{i,j}[v_j] = 1$$

then p_i is a potential prediction of $\boldsymbol{v}$. When multiple rules are matched, all corresponding predictions will be returned. The multirule matching issue could be addressed by picking one prediction with the most matches.

We provide an example to illustrate how to utilize the index $\boldsymbol{I}$ to perform SVM classification. Assume that $t = 4$, $m = 4$, and $n = 5$. Based on the extracted SVM rules in Table III, the index $\boldsymbol{I}$ could be built as shown in Fig. 3. Let the input biomedical feature be $\boldsymbol{v} = \{v_1, v_2, v_3, v_4\} = \{3, 2, 5, 4\}$. Since

TABLE III
EXAMPLE OF SVM RULES

$\boldsymbol{R}$	Conditions								$\boldsymbol{p}$
	$lb_{*,1}$	$ub_{*,1}$	$lb_{*,2}$	$ub_{*,2}$	$lb_{*,3}$	$ub_{*,3}$	$lb_{*,4}$	$ub_{*,4}$	
R_1	2	3	2	4	3	3	1	5	p_1
R_2	2	4	1	3	1	3	3	4	p_2
R_3	3	3	1	4	4	5	4	5	p_3
R_4	1	2	3	5	2	5	1	2	p_4

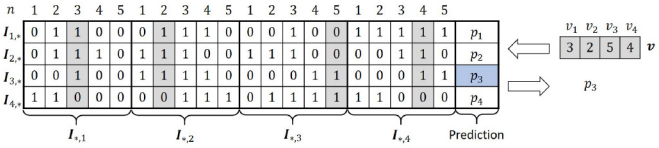

Fig. 3. Example of indices.

$\boldsymbol{v}$ satisfies the following equations, i.e.,

$$\bigwedge_{j=1}^{4} I_{1,j}[v_j] = I_{1,1}[3] \wedge I_{1,2}[2] \wedge I_{1,3}[5] \wedge I_{1,4}[4] = 0$$

$$\bigwedge_{j=1}^{4} I_{2,j}[v_j] = I_{2,1}[3] \wedge I_{2,2}[2] \wedge I_{2,3}[5] \wedge I_{2,4}[4] = 0$$

$$\bigwedge_{j=1}^{4} I_{3,j}[v_j] = I_{3,1}[3] \wedge I_{3,2}[2] \wedge I_{3,3}[5] \wedge I_{3,4}[4] = 1$$

$$\bigwedge_{j=1}^{4} I_{4,j}[v_j] = I_{4,1}[3] \wedge I_{4,2}[2] \wedge I_{4,3}[5] \wedge I_{4,4}[4] = 0$$

the prediction of $\boldsymbol{v}$ is p_3.

C. VSSVMC: *Verifiable and Secure SVM Classification*

We provide the definition of VSSVMC for cloud-based health monitoring services as follows.

*Definition 1 (*VSSVMC*):* The VSSVMC contains six polynomial-time algorithms. Namely, VSSVMC = (Init, ClfEnc, TokenGen, SecEva, Veri, Dec).

1) $\mathsf{Init}(\kappa) \rightarrow K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, K_0$. With a security parameter κ, $\mathcal{MI}$ generates $K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}$, and K_0 and sends all these parameters to authorized $\mathcal{C}$.
2) $\mathsf{ClfEnc}(K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, K_0, \boldsymbol{I}, \boldsymbol{p}) \rightarrow T_0, T_1$. First, $\mathcal{MI}$ encrypts all values in $\boldsymbol{I}$ and stores the encrypted values in the linear index T_0. Then, $\mathcal{MI}$ encrypts $\boldsymbol{p}$ as $\boldsymbol{c}$, generates $\boldsymbol{vc}$ for $\boldsymbol{c}$, and stores all these values in the encrypted linear index T_1. Finally, $\mathcal{MI}$ outsources both T_0 and T_1 to $\mathcal{CS}$.
3) $\mathsf{TokenGen}(K_{f_0}, K_{h_0}, K_{h_1}, H_0, H_1, F_0, \boldsymbol{v}) \rightarrow \boldsymbol{TK}(\boldsymbol{v})$. When $\mathcal{C}$ requests a medical decision for the m-dimensional medical feature $\boldsymbol{v}$, he generates tokens $\boldsymbol{TK}(\boldsymbol{v})$ for $\boldsymbol{v}$ and submits $\boldsymbol{TK}(\boldsymbol{v})$ to $\mathcal{CS}$.
4) $\mathsf{SecEva}(\boldsymbol{TK}(\boldsymbol{v}), T_0, T_1) \rightarrow \boldsymbol{c}(\boldsymbol{v}), \boldsymbol{PF}$. With $\boldsymbol{TK}(\boldsymbol{v})$ from $\mathcal{C}$, $\mathcal{CS}$ searches both T_0 and T_1, and produces encrypted decisions and proofs. Then, $\mathcal{CS}$ returns both $\boldsymbol{c}(\boldsymbol{v})$ and $\boldsymbol{PF}$ to $\mathcal{C}$.
5) $\mathsf{Veri}(\boldsymbol{c}(\boldsymbol{v}), \boldsymbol{PF}, K_{f_0}, K_{f_1}) \rightarrow$ ACCEPT/REJECT. After receiving $\boldsymbol{c}(\boldsymbol{v})$, $\boldsymbol{PF}$, $\mathcal{C}$ verifies whether $\mathcal{CS}$ returns correct and complete $\boldsymbol{c}(\boldsymbol{v})$ by utilizing $\boldsymbol{PF}$. Then, $\mathcal{C}$ outputs ACCEPT or REJECT for $\boldsymbol{c}(\boldsymbol{v})$.
6) $\mathsf{Dec}(\boldsymbol{c}(\boldsymbol{v}), K_0) \rightarrow \boldsymbol{p}(\boldsymbol{v})$. If $\mathcal{C}$ accepts $\boldsymbol{c}(\boldsymbol{v})$, then he decrypts $\boldsymbol{c}(\boldsymbol{v})$ and obtains $\boldsymbol{p}(\boldsymbol{v})$.

Scheme Details: The main idea of VSSVMC is to transform the SVM classification functionality to a function of conjunctively querying whether a feature vector is located in a multidimensional interval, and construct a verifiable and secure conjunctive query scheme to achieve trustworthy and secure cloud-based health monitoring services. First, a Boolean index $\boldsymbol{I}$ is constructed to represent the SVM classifier. Second, by adopting the idea of [34], $\boldsymbol{I}$ is encrypted by utilizing prf, prp, and Sym, and the encrypted results are stored in T_0 and T_1. Third, verification messages are produced and stored in both T_0 and T_1. Finally, the secure SVM evaluation and verification are achieved by searching T_0 and T_1 and verifying the returned verification messages. We provide detailed constructions of VSSVMC in Fig. 4.

As shown in Fig. 4, the workflow of verifiable and secure cloud-based health monitoring services contains six steps (algorithms). In the initialization step, $\mathcal{MI}$ produces K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, and K_0 and shares these security parameters to $\mathcal{C}$. In the classifier encryption step, $\mathcal{MI}$ utilizes prf, prp, and Sym to encrypt $\boldsymbol{I}$ and $\boldsymbol{p}$, generates $\boldsymbol{vc}$, and finally produces and outsources T_0 and T_1 to $\mathcal{CS}$. In the token generation step, $\mathcal{C}$ produces a set of tokens $\boldsymbol{TK}(\boldsymbol{v})$ for his medical feature $\boldsymbol{v}$ and submits $\boldsymbol{TK}(\boldsymbol{v})$ to $\mathcal{CS}$. In the secure evaluation step, $\mathcal{CS}$ searches T_0 and T_1 for $\boldsymbol{TK}(\boldsymbol{v})$, and returns encrypted decision $\boldsymbol{c}(\boldsymbol{v})$ and proofs $\boldsymbol{PF}$ to $\mathcal{C}$. In the verification step, $\mathcal{C}$ verifies whether $\mathcal{CS}$ returns correct $\boldsymbol{c}(\boldsymbol{v})$ by utilizing $\boldsymbol{PF}$. In the decryption step, $\mathcal{C}$ decrypts the encrypted medical decisions $\boldsymbol{c}(\boldsymbol{v})$ and obtains the decisions $\boldsymbol{p}(\boldsymbol{v})$ for $\boldsymbol{v}$.

VI. SECURITY AND VERIFIABILITY ANALYSIS

A. Security Definitions

A leakage function $\mathcal{L}$ is defined to show the information leakage of VSSVMC. With the biomedical feature $\boldsymbol{v}$ and the Boolean vector $\boldsymbol{I}$, the leakage function $\mathcal{L}(\boldsymbol{v}, \boldsymbol{I})$ could be defined as follows.

Definition 2 (Leakage Function $\mathcal{L}(\boldsymbol{v}, \boldsymbol{I})$): The leakage function involves three patterns, i.e., size patterns, search patterns, and access patterns.

1) *Size Patterns:* The size patterns are the size of T_0, T_1, and $\boldsymbol{c}(\boldsymbol{v})$, i.e., $|T_0|$, $|T_1|$, and $|\boldsymbol{c}(\boldsymbol{v})|$.
2) *Search Patterns:* The search patterns are the differences between two tokens. Namely, the search patterns denote the differences between $\boldsymbol{TK}(\boldsymbol{v})$ and $\boldsymbol{TK}(\boldsymbol{v}')$, where $\boldsymbol{v}$ and $\boldsymbol{v}'$ are two medical features.
3) *Access Patterns:* The access patterns are the mapping relations between the token and the corresponding encrypted decisions and proofs. Namely, the access patterns denote the relationship between $\boldsymbol{TK}(\boldsymbol{v})$ and the corresponding $\boldsymbol{c}(\boldsymbol{v})$.

Verifiable and Secure SVM Classification (VSSVMC)

★ **Initialization** (Init):
Inputs: κ.
Outputs: $K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, K_0$.

1: $\mathcal{MI}$: $\mathcal{MI}$ selects the security parameter κ and generates K_{f_0}, K_{f_1}, K_{h_0}, and K_{h_1} from $\{0,1\}^\kappa$. Namely,

$$K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1} \xleftarrow{\$} \{0,1\}^\kappa.$$

Then, $\mathcal{MI}$ generates the symmetric keys K_0 for Sym. Namely,

$$K_0 \leftarrow \texttt{Sym.Gen}\,(1^\kappa)\,.$$

2: $\mathcal{MI} \rightarrow \mathcal{C}$: $\mathcal{MI}$ sends K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, and K_0 to authorized $\mathcal{C}$.

★ **Classifier Encryption** (ClfEnc):
Inputs: K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, K_0, $\boldsymbol{I}$, $\boldsymbol{p}$.
Outputs: T_0, T_1.

1: $\mathcal{MI}$: For each $i \in \mathbb{Z}_t$, $j \in \mathbb{Z}_m$, $k \in \mathbb{Z}_n$, $\mathcal{MI}$ sets:

$$T_0\left[H_0\left(i||j||k\right)\right] \leftarrow F_0\left(I_{i,j}[k]||i||j||k\right).$$

2: $\mathcal{MI}$: For each $i \in \mathbb{Z}_t$, $\mathcal{MI}$ calculates:

$$c_i \leftarrow \texttt{Sym.Enc}(K_0, p_i),$$
$$vc_i \leftarrow F_1(H_1(i)||c_i),$$

where $p_i \in \boldsymbol{p}$, and sets:

$$T_1[H_1(i)] \leftarrow c_i||vc_i.$$

3: $\mathcal{MI} \rightarrow \mathcal{CS}$: $\mathcal{MI}$ outsources T_0 and T_1 to $\mathcal{CS}$.

★ **Token Generation** (TokenGen):
Inputs: K_{f_0}, K_{h_0}, K_{h_1}, H_0, H_1, F_0, $\boldsymbol{v}$.
Outputs: $\boldsymbol{TK}(\boldsymbol{v})$.

1: $\mathcal{C}$: When $\mathcal{C}$ requests a clinical decision for the medical features $\boldsymbol{v}$, $\mathcal{C}$ samples t keys for Sym. Namely,

$$K_1, K_2, \ldots, K_t \xleftarrow{\$} \{0,1\}^\kappa.$$

Then, $\mathcal{C}$ generates t tokens. Namely, $\boldsymbol{TK}(\boldsymbol{v}) = \{TK_1(\boldsymbol{v}), \ldots, TK_t(\boldsymbol{v})\}$. Each token $TK_i(\boldsymbol{v})$ involves three values and a vector. Namely, $TK_i(\boldsymbol{v}) = (\alpha_i, \beta_i, \gamma_i, \boldsymbol{L}_i)$, which are produced as follows.

$$\alpha_i = \oplus_{j\in\mathbb{Z}_m}\left(F_0(1||i||j||v_j)\right) \oplus K_i,$$
$$\beta_i = \texttt{Sym.Enc}(K_i, 0^\kappa),$$
$$\gamma_i = \texttt{Sym.Enc}(K_i, H_1(i)),$$
$$\boldsymbol{L}_i = \{H_0(i||j||v_j)\}_{j\in\mathbb{Z}_m}.$$

2: $\mathcal{C} \rightarrow \mathcal{CS}$: $\mathcal{C}$ sends $\boldsymbol{TK}(\boldsymbol{v})$ to $\mathcal{CS}$.

★ **Secure Evaluation** (SecEva):
Inputs: $\boldsymbol{TK}(\boldsymbol{v})$, T_0, T_1.
Outputs: $\boldsymbol{c}(\boldsymbol{v})$, $\boldsymbol{PF}$.

1: $\mathcal{CS}$: $\mathcal{CS}$ receives T_0 and T_1 from $\mathcal{MI}$.
2: $\mathcal{CS}$: After receiving $\boldsymbol{TK}(\boldsymbol{v})$ from $\mathcal{C}$, $\mathcal{CS}$ initialize $\boldsymbol{c}(\boldsymbol{v}) \leftarrow \emptyset$.
3: $\mathcal{CS}$: For each $TK_i(\boldsymbol{v}) \in \boldsymbol{TK}(\boldsymbol{v})$ ($i \in \mathbb{Z}_t$), $\mathcal{CS}$ calculates

$$K_i' = \oplus_{j\in\mathbb{Z}_m}\left(T_0\left[H_0(i||j||v_j)\right]\right) \oplus \alpha_i.$$

1) If $\texttt{Sym.Dec}\,(K_i', \beta_i) = 0^\kappa$, then $\mathcal{CS}$ searches $T_1\left[\texttt{Sym.Dec}(K_i', \gamma_i)\right]$ and obtains $c_i||vc_i$. Later, $\mathcal{CS}$ adds c_i to $\boldsymbol{c}(\boldsymbol{v})$ and produces $\texttt{PF}_i$, i.e.,

$$\boldsymbol{c}(\boldsymbol{v}) \leftarrow \boldsymbol{c}(\boldsymbol{v}) \cup \{c_i\},$$
$$\texttt{PF}_i \leftarrow K_i' \,||\, vc_i.$$

2) Else if $\texttt{Sym.Dec}\,(K_i', \beta_i) \neq 0^\kappa$, then $\mathcal{CS}$ adds $\emptyset$ to $\boldsymbol{c}(\boldsymbol{v})$ and produces $\texttt{PF}_i$, i.e.,

$$\boldsymbol{c}(\boldsymbol{v}) \leftarrow \boldsymbol{c}(\boldsymbol{v}) \cup \emptyset,$$
$$\texttt{PF}_i \leftarrow \{T_0[H_0(i||j||v_j)]\}_{j\in\mathbb{Z}_m}.$$

4: $\mathcal{CS} \rightarrow \mathcal{C}$: $\mathcal{CS}$ returns $\boldsymbol{c}(\boldsymbol{v})$ and $\boldsymbol{PF} = \{\texttt{PF}_1, \ldots, \texttt{PF}_t\}$ to $\mathcal{C}$.

★ **Verification** (Veri):
Inputs: $\boldsymbol{c}(\boldsymbol{v})$, $\boldsymbol{PF}$, K_{f_0}, K_{f_1}.
Outputs: ACCEPT/REJECT.

1: $\mathcal{C}$: $\mathcal{C}$ receives $\boldsymbol{c}(\boldsymbol{v})$ and $\boldsymbol{PF}$ from $\mathcal{CS}$.
2: $\mathcal{C}$: For each $i \in \mathbb{Z}_t$, $\mathcal{C}$ may face to two cases:

1) If $c_i \in \boldsymbol{c}(\boldsymbol{v})$, then $\mathcal{C}$ verifies K_i' and vc_i, i.e., if $K_i \neq K_i'$ or $F_1(H_1(i)||c_i) \neq vc_i$, then outputs REJECT.
2) Else if $c_i \notin \boldsymbol{c}(\boldsymbol{v})$, then $\mathcal{C}$ verifies $\texttt{PF}_i$, i.e., if $\forall v_j \in \boldsymbol{v}$, $F_0(0||i||j||v_j) \neq T_0[H_0(i||j||v_j)]$, then outputs REJECT.

3: $\mathcal{C}$: If the algorithm didn't outputs a REJECT, then $\mathcal{C}$ accepts $\boldsymbol{c}(\boldsymbol{v})$ and outputs ACCEPT.

★ **Decryption** (Dec):
Inputs: $\boldsymbol{c}(\boldsymbol{v})$, K_0.
Outputs: $\boldsymbol{p}(\boldsymbol{v})$.

1: $\mathcal{C}$: If $\mathcal{C}$ accepts $\boldsymbol{c}(\boldsymbol{v})$, then for each $c_i \in \boldsymbol{c}(\boldsymbol{v})$, $\mathcal{C}$ calculates and obtains

$$\boldsymbol{p}(\boldsymbol{v}) = \{p_i \mid p_i = \texttt{Sym.Dec}(K_0, c_i), c_i \in \boldsymbol{c}(\boldsymbol{v})\}.$$

Fig. 4. Detail construction of VSSVMC for cloud-based health monitoring systems.

The leakage function $\mathcal{L}(\boldsymbol{v}, \boldsymbol{I})$ indicates the default information revealed in most of the searchable symmetric encryption [30], [34] and SSE-based machine learning classification schemes [8], [23]. With $\mathcal{L}(\boldsymbol{v}, \boldsymbol{I})$, the adaptive $\mathcal{L}$-security definition is given as follows.

Definition 3 (Adaptive $\mathcal{L}$-Security): Let Ψ be a VSSVMC with six algorithms. Namely, Ψ = (Init, ClfEnc, TokenGen, SecEva, Veri, Dec). Let $\mathcal{A} = (\mathcal{A}^0, \mathcal{A}^1, \ldots, \mathcal{A}^q)$ be an adversary, where $q \in \mathbb{N}$. Let $\mathcal{S} = (\mathcal{S}^0, \mathcal{S}^1, \ldots, \mathcal{S}^q)$ be a simulator, where $q \in \mathbb{N}$. Let $\boldsymbol{v}^1, \boldsymbol{v}^2, \ldots, \boldsymbol{v}^q$ be q medical features produced by $\mathcal{A}$. We define the real-world experiment [i.e., $\boldsymbol{Real}_\Psi^{\mathcal{A}}(1^\kappa)$] and the simulation [i.e., $\boldsymbol{Sim}_{\mathcal{L},\mathcal{S}}^{\mathcal{A}}(1^\kappa)$] as follows.

1) $\boldsymbol{Real}_\Psi^{\mathcal{A}}(1^\kappa)$: At round 0, the challenger randomly produces K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, and K_0 by invoking Init(κ). Then, $\mathcal{A}^0$ generates an SVM classifier *SVM*,

constructs the Boolean vector $\boldsymbol{I}$, and produces the corresponding predictions $\boldsymbol{p}$. Afterward, the challenger utilizes $\mathsf{ClfEnc}(K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, K_0, \boldsymbol{I}, \boldsymbol{p})$ to produce two tables T_0 and T_1, and outsources both T_0 and T_1 to $\mathcal{A}$. Later, $\mathcal{A}$ makes q classification requests, at round r ($1 \leq r \leq q$): $\mathcal{A}^r$ reviews $\boldsymbol{v}^1, \ldots, \boldsymbol{v}^{r-1}$ and the corresponding predictions $\boldsymbol{c}(\boldsymbol{v}^1), \ldots, \boldsymbol{c}(\boldsymbol{v}^{r-1})$. Then, $\mathcal{A}^r$ produces $\boldsymbol{v}^r$ adaptively. After that, the challenger generates $\boldsymbol{TK}(\boldsymbol{v}^r)$ by invoking $\mathsf{TokenGen}(K_{f_0}, K_{h_0}, K_{h_1}, H_0, H_1, F_0, \boldsymbol{v})$, and sends $\boldsymbol{TK}(\boldsymbol{v}^r)$ to $\mathcal{A}^r$. Finally, $\mathcal{A}^r$ searches T_0 and T_1 by utilizing $\boldsymbol{TK}(\boldsymbol{v}^r)$. After q rounds interactions, $\mathcal{A}$ outputs a bit as the output of the real-world experiment.

2) $\boldsymbol{Sim}^{\mathcal{A}}_{\mathcal{L},\mathcal{S}}(1^\kappa)$*:* At round 0, with $\mathcal{L}(\boldsymbol{v}, \boldsymbol{I})$, $\mathcal{S}_0$ produces two tables T_0^* and T_1^* with random strings and outsources both T_0^* and T_1^* to $\mathcal{A}$. Later, $\mathcal{A}$ makes q classification requests, at round r ($1 \leq r \leq q$): $\mathcal{A}^r$ reviews $\boldsymbol{v}^1, \ldots, \boldsymbol{v}^{r-1}$ and the corresponding predictions $\boldsymbol{c}(\boldsymbol{v}^1), \ldots, \boldsymbol{c}(\boldsymbol{v}^{r-1})$. Then, $\mathcal{A}^r$ produces $\boldsymbol{v}^r$ adaptively. With $\mathcal{L}(\boldsymbol{v}, \boldsymbol{I})$, $\mathcal{S}_r$ produces appropriate $\boldsymbol{TK}(\boldsymbol{v}^r)^*$. Finally, $\mathcal{A}^r$ searches T_0^* and T_1^* by utilizing $\boldsymbol{TK}(\boldsymbol{v}^r)^*$. After q rounds interactions, $\mathcal{A}$ outputs a bit as the output of the simulation experiment.

We define that Ψ is adaptively $\mathcal{L}$-secure if for all polynomial size adversaries $\mathcal{A}$, there exists a simulator $\mathcal{S}$, such that the probability of the difference between the output of $\mathbf{Real}^{\mathcal{A}}_{\Psi}(1^\kappa)$ and $\mathbf{Sim}^{\mathcal{A}}_{\mathcal{L},\mathcal{S}}(1^\kappa)$ is negligible. Namely

$$\left| Pr\left[\mathbf{Real}^{\mathcal{A}}_{\Psi}(1^\kappa) = 1\right] - Pr\left[\mathbf{Sim}^{\mathcal{A}}_{\mathcal{L},\mathcal{S}}(1^\kappa) = 1\right]\right| \leq \mathtt{negl}(\kappa).$$

B. Security Proofs

Theorem 1: VSSVMC is adaptively $\mathcal{L}$-secure if F_0 and F_1 are $\mathtt{prf}$, H_0 and H_1 are $\mathtt{prp}$, and $\mathtt{Sym}$ is an IND-CPA secure symmetric key encryption.

Proof: We produce a simulator $\mathcal{S} = (\mathcal{S}^0, \ldots, \mathcal{S}^q)$ such that for any polynomial-size adversary $\mathcal{A} = (\mathcal{A}^0, \ldots, \mathcal{A}^q)$, the outputs of $\mathbf{Real}^{\mathcal{A}}_{\Psi}(1^\kappa)$ and $\mathbf{Sim}^{\mathcal{A}}_{\mathcal{L},\mathcal{S}}(1^\kappa)$ are computationally indistinguishable. $\mathcal{S} = (\mathcal{S}^0, \ldots, \mathcal{S}^q)$ generates T_0^*, T_1^*, and $\boldsymbol{TK}(\boldsymbol{v}^r)^*$ adaptively as follows.

$\mathcal{S}^0$*:* With the size pattern in $\mathcal{L}(\boldsymbol{v}, I)$, $\mathcal{S}^0$ obtains t, m, and n. Then, $\mathcal{S}^0$ produces tmn κ-bit random strings as T_0^*. Namely, T_0^* is a linear vector with tmn elements, which are κ-bit random strings. After that, $\mathcal{S}^0$ randomly generates t 2κ-bit strings as T_1^*. Namely, T_1^* is a linear vector with t elements, which are 2κ-bit random strings. Finally, $\mathcal{S}^0$ outsources both T_0^* and T_1^* to $\mathcal{A}^0$. With all but negligible probability, $\mathcal{A}$ cannot recover K_{f_0}. As a result, for each $i \in \mathbb{Z}_t$, $j \in \mathbb{Z}_m$, and $k \in \mathbb{Z}_n$, distinguishing the pseudorandom output of $F_0(I_{i,j}[k]||i||j||k)$ in T_0 from a κ-bit random string in T_0^* is difficult for $\mathcal{A}$, if F_0 is a $\mathtt{prf}$. Similarly, with all but negligible probability, $\mathcal{A}$ cannot recover both K_{f_1} and K_0. Thus, for each $i \in \mathbb{Z}_t$, it is hard for $\mathcal{A}$ to distinguish the concatenation of $\mathtt{Sym.Enc}(K_0, p_i)$ and $F_1(H_1(i)||c_i)$ from a 2κ-bit random string in T_1^*, if F_1 is a $\mathtt{prf}$ and $\mathtt{Sym}$ is an IND-CPA secure symmetric key encryption. Therefore, both T_0^* and T_1^* are computationally indistinguishable from T_0 and T_1, respectively.

$\mathcal{S}^r$ *(For $1 \leq r \leq q$):* By utilizing the search pattern in $\mathcal{L}(\boldsymbol{v}, I)$, $\mathcal{S}^r$ searches whether the medical feature $\boldsymbol{v}^r$ has been submitted before. We have the following two cases.

a) The medical feature $\boldsymbol{v}^r$ has totally been submitted before, i.e., there exists $\boldsymbol{v}^u$, such that $\forall\ v_j^u \in \boldsymbol{v}^u$, $v_j^u = v_j^r$, where $1 \leq u < r$ and $j \in \mathbb{Z}_n$. With the access pattern in $\mathcal{L}(\boldsymbol{v}, I)$, $\mathcal{S}^r$ sends $\boldsymbol{TK}(\boldsymbol{v}^u)$ as an appropriate token for $\boldsymbol{TK}(\boldsymbol{v}^r)^*$ to $\mathcal{A}^r$.

b) The medical feature $\boldsymbol{v}^r$ has partially or never been submitted before. $\mathcal{S}^r$ produces $\boldsymbol{TK}(\boldsymbol{v}^r)^*$ as follows. First, with the size pattern $|\boldsymbol{c}(\boldsymbol{v}^r)|$ in $\mathcal{L}(\boldsymbol{v}, I)$, $\mathcal{S}^r$ randomly selects $|\boldsymbol{c}(\boldsymbol{v}^r)|$ values in $\mathbb{Z}_t$, i.e., $\{x_1, \ldots, x_{|\boldsymbol{c}(\boldsymbol{v}^r)|}\}$. Second, for each $x \in \{x_1, \ldots, x_{|\boldsymbol{c}(\boldsymbol{v}^r)|}\}$, $\mathcal{S}^r$ generates $TK_x^*(\boldsymbol{v}^r) = (\alpha_x^{r*}, \beta_x^{r*}, \gamma_x^{r*}, \boldsymbol{L}_x^{r*})$ as follows.

i) By utilizing the search pattern in $\mathcal{L}(\boldsymbol{v}, \boldsymbol{I})$, $\mathcal{S}^r$ searches whether v_j^r has been submitted before, where $j \in \mathbb{Z}_m$. For each $j \in \mathbb{Z}_m$, if $\exists v_j^u = v_j^r$, where $1 \leq u < r$, $\mathcal{S}^r$ chooses the same locations of v_j^u in T_0^* as $H_0(x||j||v_j^r)^*$ for v_j^r. Otherwise, $\mathcal{S}^r$ randomly chooses a location that has not been chosen in T_0^* as $H_0(x||j||v_j^r)^*$. Finally, $\mathcal{S}^r$ generates $\boldsymbol{L}_x^{r*} = \{H_0(x||j||v_j^r)^*\}_{j \in \mathbb{Z}_m}$.

ii) $\mathcal{S}^r$ randomly produces a $\mathtt{Sym}$ key K_x^*. Then, $\mathcal{S}^r$ calculates

$$\alpha_x^{r*} = \oplus_{j \in \mathbb{Z}_m}\left(T_0^*\left[H_0\left(x||j||v_j^r\right)^*\right]\right) \oplus K_x^*.$$
$$\beta_x^{r*} = \mathtt{Sym.Enc}\left(K_x^*, 0^\kappa\right)$$
$$\gamma_x^{r*} = \mathtt{Sym.Enc}\left(K_x^*, x\right).$$

Third, $\mathcal{S}^r$ produces $TK_x^*(\boldsymbol{v}^r) = (\alpha_x^{r*}, \beta_x^{r*}, \gamma_x^{r*}, \boldsymbol{L}_x^{r*})$ for each $x \in \mathbb{Z}_t$ and $x \notin \{x_1, \ldots, x_{|\boldsymbol{c}(\boldsymbol{v}^r)|}\}$ as follows.

i) By utilizing the search pattern in $\mathcal{L}(\boldsymbol{v}, \boldsymbol{I})$, $\mathcal{S}^r$ searches whether v_j^r has been submitted before, where $j \in \mathbb{Z}_m$. For each $j \in \mathbb{Z}_m$, if $\exists v_j^u = v_j^r$, where $1 \leq u < r$, $\mathcal{S}^r$ chooses the same locations of v_j^u in T_0^* as $H_0(x||j||v_j^r)^*$ for v_j^r. Otherwise, $\mathcal{S}^r$ randomly chooses a location that has not been chosen in T_0^* as $H_0(x||j||v_j^r)^*$. Finally, $\mathcal{S}^r$ generates $\boldsymbol{L}_x^{r*} = \{H_0(x||j||v_j^r)^*\}_{j \in \mathbb{Z}_m}$.

ii) $\mathcal{S}^r$ randomly chooses three κ-bit string as α_x^{r*}, β_x^{r*}, and γ_x^{r*}, i.e.,

$$\alpha_x^{r*}, \beta_x^{r*}, \gamma_x^{r*} \xleftarrow{\$} \{0,1\}^\kappa.$$

Finally, $\mathcal{S}^r$ returns $\boldsymbol{TK}(\boldsymbol{v}^r)^*$ to $\mathcal{A}^r$. $\mathcal{A}^r$ searches T_0^* and T_1^* for predictions of $\boldsymbol{v}^r$.

With all but negligible probability, $\mathcal{A}^r$ cannot recover K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, and K_0. If H_0 is a $\mathtt{prp}$, $\mathcal{A}^r$ cannot distinguish $\boldsymbol{L}_x^{r*}$ from $\boldsymbol{L}_x^r$ because distinguishing a randomly selected permutation from an output of $\mathtt{prp}$ is hard, where $x \in \mathbb{Z}_t$. Meanwhile, if F_0 is a $\mathtt{prf}$, $\mathcal{A}^r$ cannot distinguish α_x^{r*} from α_x^r because distinguishing a random string from an output of $\mathtt{prf}$ is hard,

where $x \in \mathbb{Z}_t$. Furthermore, if Sym is IND-CPA secure, $\mathcal{A}^r$ can hardly distinguish β_x^{r*} and γ_x^{r*} from β_x^r and γ_x^r, respectively, because it is hard to distinguish the ciphertext of Sym from a random string, where $x \in \mathbb{Z}_t$. Therefore, for each $x \in \mathbb{Z}_t$, it is hard for $\mathcal{A}^r$ to distinguish $TK_x^*(\boldsymbol{v}^r)$ from $TK_x(\boldsymbol{v}^r)$, which further indicates that $\mathcal{A}^r$ cannot distinguish $\boldsymbol{TK}(\boldsymbol{v}^r)^*$ from $\boldsymbol{TK}(\boldsymbol{v}^r)$. Since both T_0^* and T_1^* are computationally indistinguishable from T_0 and T_1, respectively, $\boldsymbol{c}(\boldsymbol{v}^r)$ are computationally indistinguishable from $\boldsymbol{c}(\boldsymbol{v}^r)^*$.

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

C. Verifiability Definitions

In cloud-based health monitoring services, a malicious $\mathcal{CS}$ may return a fraction of SVM classification results or manipulate the SVM classification results deliberately. To ensure the reliability of cloud-based health monitoring services, it is necessary to enable $\mathcal{C}$ to verify whether $\mathcal{CS}$ faithfully executes the SVM classification processes and returns correct and complete medical decisions produced by the SVM classifier. Both the correctness and completeness of VSSVMC are defined in Definitions 4 and 5, respectively.

Definition 4 (Correctness): For a medical feature $\boldsymbol{v}$ and a Boolean index $\boldsymbol{I}$ (i.e., the SVM classifier), the encrypted decision set $\boldsymbol{c}(\boldsymbol{v})$ is returned. If each encrypted decision $c_i \in \boldsymbol{c}(\boldsymbol{v})$ is a matched encrypted decision of $\boldsymbol{v}$, then $\boldsymbol{c}(\boldsymbol{v})$ is correct.

Definition 5 (Completeness): For a medical feature $\boldsymbol{v}$ and a Boolean index $\boldsymbol{I}$ (i.e., the SVM classifier), the encrypted decision set $\boldsymbol{c}(\boldsymbol{v})$ is returned. If all matched decision c_i are in $\boldsymbol{c}(\boldsymbol{v})$ and returned, then $\boldsymbol{c}(\boldsymbol{v})$ is complete.

In VSSVMC, a malicious adversary ($\mathcal{CS}$) may return a false (incorrect or incomplete) result to $\mathcal{C}$, i.e., $\boldsymbol{c}(\boldsymbol{v})^* \neq \boldsymbol{c}(\boldsymbol{v})$. Furthermore, the adversary might answer a wrong pair $(\boldsymbol{c}(\boldsymbol{v})^*, \boldsymbol{PF}^*)$ to raise the probability of cheating $\mathcal{C}$, i.e., $(\boldsymbol{c}(\boldsymbol{v})^*, \boldsymbol{PF}^*) \neq (\boldsymbol{c}(\boldsymbol{v}), \boldsymbol{PF})$. Similar to the strong reliability definition in [36], the verifiability protects the VSSVMC against the aforementioned malicious behaviors. The definition of verifiability is given as follows.

Definition 6 (Verifiability): Let Ψ be a VSSVMC, i.e., $\Psi = (\mathsf{Init}, \mathsf{ClfEnc}, \mathsf{TokenGen}, \mathsf{SecEva}, \mathsf{Veri}, \mathsf{Dec})$. Let $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ be a malicious adversary, where $\mathcal{B}_1$ and $\mathcal{B}_2$ are supposed to be able to communicate freely. Let $\boldsymbol{v}^1, \boldsymbol{v}^2, \ldots, \boldsymbol{v}^q$ be q medical features produced by $\mathcal{B}_1$, where $q \in \mathbb{N}$. The verifiability game, which is played by $\mathcal{B}$ and a challenger, is defined as follows.

Game$_{\Psi,veri}^{\mathcal{B}}(1^\kappa)$*:*

1) $\mathcal{B}_1$ produces $(\boldsymbol{I}, \boldsymbol{p})$ and sends them to the challenger.
2) The challenger obtains K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, and K_0 by invoking $\mathsf{Init}(\kappa)$. Then, the challenger generates T_0 and T_1 by utilizing $\mathsf{ClfEnc}(K_{f_0}, K_{f_1}, K_{h_0}, K_{h_1}, K_0, \boldsymbol{I}, \boldsymbol{p})$. Finally, the challenger outsources both T_0 and T_1 to $\mathcal{B}_2$.
3) The challenger and $\mathcal{B}$ engage in q rounds. At round r $(1 \leq r \leq q)$:
   a) $\mathcal{B}_1$ selects and sends $\boldsymbol{v}^r$ to the challenger.
   b) The challenger produces the token $\boldsymbol{TK}(\boldsymbol{v}^r)$ by invoking $\mathsf{TokenGen}(K_{f_0}, K_{h_0}, K_{h_1}, H_0, H_1, F_0, \boldsymbol{v})$, and sends $\boldsymbol{TK}(\boldsymbol{v}^r)$ to $\mathcal{B}_2$.
   c) $\mathcal{B}_2$ returns $(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*})$ to the challenger.
   d) The challenger calculates

$$\texttt{ACCEPT/REJECT} \leftarrow \mathsf{Veri}\big(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*}, K_{f_0}, K_{f_1}\big).$$

   If the output is ACCEPT, the challenger sends $\boldsymbol{p}(\boldsymbol{v}^r)^* \leftarrow \mathsf{Dec}(\boldsymbol{c}(\boldsymbol{v}^r)^*, K_0)$ to $\mathcal{B}_1$. Otherwise, the challenger returns REJECT to $\mathcal{B}_1$.
4) If there exists an r, such that both $(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*}) \neq (\boldsymbol{c}(\boldsymbol{v}^r), \boldsymbol{PF}^r)$ and $\mathsf{Veri}(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*}, K_{f_0}, K_{f_1}) = \texttt{ACCEPT}$ holds simultaneously, the game outputs 1. Otherwise, the game outputs 0.

We say that $\mathcal{B}$ wins the game if $\mathbf{Game}_{\Psi,\mathsf{veri}}^{\mathcal{B}}(1^\kappa) = 1$. We say that a VSSVMC holds verifiability if for any probabilistic polynomial time adversary $\mathcal{B}$, the probability of $\mathcal{B}$ wins $\mathbf{Game}_{\Psi,\mathsf{veri}}^{\mathcal{B}}(1^\kappa)$ is negligible, i.e.,

$$Pr\Big[\boldsymbol{Game}_{\Psi,\mathsf{veri}}^{\mathcal{B}}\big(1^\kappa\big) = 1\Big] = \mathtt{negl}(\kappa).$$

D. Verifiability Proofs

Theorem 2: VSSVMC ensures verifiability if F_0 and F_1 are prf, *Sym* is an IND-CPA secure symmetric key encryption.

Proof: Suppose that there exists an probabilistic polynomial time adversary $\mathcal{B}$, who can break the verifiability of VSSVMC. Namely, the probability of $\mathcal{B}$ wins $\mathbf{Game}_{\Psi,\mathsf{veri}}^{\mathcal{B}}(1^\kappa)$ is not negligible. Assume that $\mathcal{B}_2$ returns $(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*})$, such that

$$\big(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*}\big) \neq \big(\boldsymbol{c}(\boldsymbol{v}^r), \boldsymbol{PF}^r\big)$$
$$\mathsf{Veri}\big(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*}, K_{f_0}, K_{f_1}\big) = \texttt{ACCEPT}$$

where $1 \leq r \leq q$. For each $(c_i^{r*}, \mathtt{PF}_i^{r*})$, we have the following two cases, where $i \in \mathbb{Z}_t$.

1) If $c_i^{r*} \in \boldsymbol{c}(\boldsymbol{v})^{r*}$, then $\mathtt{PF}_i^{r*} = K_i'^{r*}||vc_i^{r*}$. Otherwise, $\mathsf{Veri}(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*}, K_{f_0}, K_{f_1})$ outputs REJECT. If $c_i^{r*} \neq c_i^r$, there are two probabilities.
   a) $\mathcal{B}_2$ adaptively selects $(c_i^u, \mathtt{PF}_i^u)$ as $(c_i^{r*}, \mathtt{PF}_i^{r*})$, where $1 \leq u \leq r$, $c_i^u \in T_0$, and

$$\mathsf{Veri}\big(\boldsymbol{c}\big(\boldsymbol{v}^u\big), \boldsymbol{PF}^u, K_{f_0}, K_{f_1}\big) = \texttt{ACCEPT}.$$

   For each $i \in \mathbb{Z}_t$, $K_i'^r$ is a one-time produced key of Sym. With negligible probability, $\mathcal{B}_2$ can recover $K_i'^r$ and $\mathsf{Veri}(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*}, K_{f_0}, K_{f_1})$ outputs ACCEPT. Otherwise, $\mathcal{B}_2$ can forge a random key $K_i'^{r*}$, such that $K_i'^{r*} = K_i'^r$.
   b) $\mathcal{B}_2$ forges $(c_i^{r*}, \mathtt{PF}_i^{r*})$, such that $c_i^{r*} \notin T_0$. For each $i \in \mathbb{Z}_t$, vc_i^r is an output of prf. With negligible probability, $\mathcal{B}_2$ can recover K_{f_1}, produce $F_1(H_1(i)||c_i^{r*}) = F_1(H_1(i)||c_i^r)$, and let $\mathsf{Veri}(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*}, K_{f_0}, K_{f_1})$ outputs ACCEPT. Otherwise, $\mathcal{B}_2$ can break prf and forge $vc_i^{r*} = F_1(H_1(i)||c_i^r)$.
2) If $c_i^{r*} \notin \boldsymbol{c}(\boldsymbol{v})^{r*}$, then

$$\begin{aligned}\mathtt{PF}_i^{r*} &= \Big\{T_0\big[H_0\big(i||j||v_j\big)\big]\Big\}_{j\in\mathbb{Z}_m}^{r*} \\ &= \big\{F_0(I_{i,j}[v_j]||i||j||v_j)\big\}_{j\in\mathbb{Z}_m}^{r*}.\end{aligned}$$

   Otherwise, $\mathsf{Veri}(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*}, K_{f_0}, K_{f_1})$ outputs REJECT. For each $i \in \mathbb{Z}_t$ and $j \in \mathbb{Z}_m$, $F_0(I_{i,j}[v_j]||i||j||v_j)^r$ is an output of prf. With

TABLE IV
PARAMETERS FOR PERFORMANCE ANALYSIS

Notation	Description
$C_{\mathtt{gen}}$	Computational costs of Sym key generation.
$C_{\mathtt{enc}}$	Computational costs of Sym encryption.
$C_{\mathtt{dec}}$	Computational costs of Sym decryption.
$C'_{\mathtt{prf}}$	Computational costs of prf key generation.
$C_{\mathtt{prf}}$	Computational costs of prf computation.
$C'_{\mathtt{prp}}$	Computational costs of prp key generation.
$C_{\mathtt{prp}}$	Computational costs of prp computation.
$C_{\mathtt{xor}}$	Computational costs of excl sive-or operation.
$C_{\mathtt{and}}$	Computational costs of bitwise-AND operation.
$S_{\mathtt{Sym}}$	Size of Sym ciphertexts.
$S_{\mathtt{prf}}$	Size of prf outputs.

negligible probability, $\mathcal{B}_2$ can recover K_{f_0}, produce a $F_0(0||i||j||v_j)^{r*}$, and let $\mathsf{Veri}(\boldsymbol{c}(\boldsymbol{v}^r)^*, \boldsymbol{PF}^{r*}, K_{f_0}, K_{f_1})$ outputs ACCEPT. Otherwise, $\mathcal{B}_2$ can break prf and forge a $F_0(0||i||j||v_j)^{r*} = F_0(0||i||j||v_j)^r$.

Therefore, $\mathcal{B}$ can only win $\mathbf{Game}^{\mathcal{B}}_{\Psi,\mathsf{veri}}(1^\kappa)$ in a negligible probability. ■

In summary, the VSSVMC protects both the medical decision model and medical data against the malicious $\mathcal{CS}$. We have provided formal security and verifiability proofs to demonstrate that VSSVMC captures the adaptive $\mathcal{L}$-security definition and ensures the verifiability.

VII. PERFORMANCE ANALYSIS AND EVALUATIONS

A. Performance Analysis

Let n, m, and t be the normalized interval of medical features, the number of dimensions of medical feature, and the number of extracted rules, respectively. We utilize $|\boldsymbol{c}(\boldsymbol{v})|$ to denote the number of matched encrypted decisions for an input token. We summarize a list of parameters that needed in performance analysis in Table IV. In Table V, we analyze the performance of VSSVMC in terms of computational costs, storage costs, and communication costs. We also compare the aforementioned performance properties of VSSVMC with the latest SSE-based secure SVM classification scheme [23] (*LQNLS20*). Since *LQNLS20* is constructed based on the Bloom filters, we utilize d and k as the number of enumerated values and the number of hash functions in each Bloom filter, respectively. Note that d is a value that satisfies $1 \leq d \leq n$.

Computational Costs: In cloud-based health monitoring services, since the SVM classifier training and rule extraction of the SVM classifier are completed by $\mathcal{MI}$, parameters t, m, and n are constants. Namely, the computational costs of Init, ClfEnc, TokenGen, and SecEva of VSSVMC are constants. Meanwhile, the computational costs of Veri and Dec of VSSVMC are depended on $|\boldsymbol{c}(\boldsymbol{v})|$. Table V illustrates the computational cost of each algorithm in VSSVMC, which shows that the computational costs of each algorithm of VSSVMC is similar to that of *LQNLS20*.

Communication and Storage Costs: We analyze the storage and communication costs for VSSVMC by calculating the size of outsourced indices, the encrypted tokens, and the encrypted decisions. As shown in Table V, both the storage costs and the communication costs of submitting a medical decision request of VSSVMC are constants once the SVM classifier is pretrained. Meanwhile, the communication cost of receiving the encrypted medical decisions is depended on $|\boldsymbol{c}(\boldsymbol{v})|$. The analysis results show that both the storage costs and the communication costs of VSSVMC are similar to that of *LQNLS20*.

Compared with the scheme in [23] (*LQNLS20*), VSSVMC ensures not only the verifiability of medical decisions but also the accuracy of SVM classification for health monitoring services. Note that the verifiability ensures the VSSVMC secure against malicious adversaries, which is an essential functionality to build a trust and secure health monitoring systems. Furthermore, without utilizing the Bloom filter technique, VSSVMC will not introduce a false positive to the encrypted indices for SVM classification, which ensures the accuracy of medical decisions. Our experimental evaluations will further compare the performance differences between VSSVMC and *LQNLS20* in a real-world medical data set.

B. Experimental Settings

All codes are developed in C++ based on OpenSSL.[1] All experiments are run on a 64-bit VMware Workstation (running Ubuntu 18.04) with 8-GB RAM and an Intel Core i7-8850H processor (2.60 GHz). The prf and the Sym are implemented by HMAC-256 and AES-CBC-256, respectively. We implement prp by utilizing prf.

Data Sets and Extracted Rules: The Breast-Cancer-Wisconsin[2] data set, which contains 683 nonmissing medical records, is utilized to train the medical decision model (also known as, the SVM classifier). All records in the tested data set contain nine discrete features, whose values are in the interval $[1, 10]$. By extracting rules from a pretrained SVM classifier and optimizing the extracted rules, we produce 16 medical decision rules in Fig. 5, which contain both benign decision rules and malignant decision rules. We evaluate the true positive, false positive, false negative, true negative, and the decision accuracy of different SVM classifiers (or extracted SVM rules), i.e., the rules in Fig. 5, the original SVM classifier, and the rules in [23]. As shown in Table VI, the extracted rules in Fig. 5 achieve a better performance than the rules in [23]. Therefore, the extracted rules in Fig. 5 are utilized to represent the medical decision model for performance evaluation.

Baselines: We show the performance advantages of VSSVMC by comparing the computational, communication, and storage costs with some existing secure SVM classification schemes.

1) *Scheme in [16] (BPTG15): BPTG15* is a well-known secure SVM classification scheme that focuses on a server–client two-party system model. Similar to most of the secure SVM classification schemes [18], *BPTG15* is

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

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

TABLE V
PERFORMANCE PROPERTIES OF SECURE SVM CLASSIFICATION SCHEMES

Performance Properties	VSSVMC	The Scheme in [23] (**LQNLS20**)
Comp. Costs (Init)	$2 \cdot (C'_{\mathtt{prp}} + C'_{\mathtt{prf}}) + C_{\mathtt{gen}}$	$t \cdot m \cdot (k+1) \cdot C'_{\mathtt{prf}} + t \cdot C'_{\mathtt{gen}} + 2 \cdot C'_{\mathtt{prp}}$
Comp. Costs (ClfEnc)	$t \cdot m \cdot n \cdot (C_{\mathtt{prf}} + C_{\mathtt{prp}}) + t \cdot (C_{\mathtt{enc}} + C_{\mathtt{prf}} + 2 \cdot C_{\mathtt{prp}})$	$t \cdot m \cdot d \cdot k \cdot (2 \cdot C_{\mathtt{prf}} + C_{\mathtt{prp}}) + t \cdot (C_{\mathtt{enc}} + C_{\mathtt{prp}})$
Comp. Costs (TokenGen)	$t \cdot (C_{\mathtt{gen}} + (m+1) \cdot C_{\mathtt{xor}} + 2 \cdot C_{\mathtt{enc}} + C_{\mathtt{prp}} + m \cdot C_{\mathtt{prf}})$	$t \cdot m \cdot k \cdot (2 \cdot C_{\mathtt{prf}} + C_{\mathtt{prp}}) + t \cdot C_{\mathtt{prp}}$
Comp. Costs (SecEva)	$t \cdot (m+1) * C_{\mathtt{xor}} + t \cdot C_{\mathtt{dec}} + \lvert\boldsymbol{c}(\boldsymbol{v})\rvert \cdot C_{\mathtt{dec}}$	$t \cdot m \cdot k \cdot C_{\mathtt{and}}$
Comp. Costs (Veri)	$\lvert\boldsymbol{c}(\boldsymbol{v})\rvert \cdot C_{\mathtt{prf}} + (t - \lvert\boldsymbol{c}(\boldsymbol{v})\rvert) \cdot m \cdot C_{\mathtt{prf}}$	N/A
Comp. Costs (Dec)	$\lvert\boldsymbol{c}(\boldsymbol{v})\rvert \cdot C_{\mathtt{dec}}$	$\lvert\boldsymbol{c}(\boldsymbol{v})\rvert \cdot C_{\mathtt{dec}}$
Storage Costs (Indexes)	$t \cdot m \cdot n \cdot S_{\mathtt{prf}} + t \cdot (S_{\mathtt{Sym}} + S_{\mathtt{prf}})$	$t \cdot m \cdot k \cdot d \cdot \log_2(e) + t \cdot S_{\mathtt{Sym}}$
Comm. Costs (Tokens)	$t \cdot (S_{\mathtt{prf}} + 2 \cdot S_{\mathtt{Sym}} + m \cdot \log_2(t \cdot m \cdot n))$	$t \cdot m \cdot k \cdot \log_2(t \cdot m \cdot k \cdot d \cdot \log_2(e)) + t \cdot \log_2(t)$
Comm. Costs (Decisions)	$\lvert\boldsymbol{c}(\boldsymbol{v})\rvert \cdot (2 \cdot S_{\mathtt{sym}} + S_{\mathtt{prf}}) + (t - \lvert\boldsymbol{c}(\boldsymbol{v})\rvert) \cdot m \cdot S_{\mathtt{prf}}$	$\lvert\boldsymbol{c}(\boldsymbol{v})\rvert \cdot S_{\mathtt{sym}}$

R_1: $f_1 \in [1,7]$, $f_2 \in [1,9]$, $f_3 \in [1,9]$, $f_4 \in [1,7]$, $f_5 \in [1,6]$, $f_6 \in [1,6]$, $f_7 \in [1,9]$, $f_8 \in [1,6]$, $f_9 \in [1,6]$, Benign;
R_2: $f_1 \in [1,7]$, $f_2 \in [1,9]$, $f_3 \in [1,9]$, $f_4 \in [1,7]$, $f_5 \in [1,6]$, $f_6 \in [1,6]$, $f_7 \in [1,8]$, $f_8 \in [1,6]$, $f_9 \in [1,7]$, Benign;
R_3: $f_1 \in [1,7]$, $f_2 \in [1,9]$, $f_3 \in [1,9]$, $f_4 \in [1,7]$, $f_5 \in [1,6]$, $f_6 \in [1,7]$, $f_7 \in [1,6]$, $f_8 \in [1,7]$, $f_9 \in [1,2]$, Benign;
R_4: $f_1 \in [1,8]$, $f_2 \in [1,4]$, $f_3 \in [1,6]$, $f_4 \in [1,4]$, $f_5 \in [1,10]$, $f_6 \in [1,10]$, $f_7 \in [1,5]$, $f_8 \in [1,8]$, $f_9 \in [1,8]$, Benign;
R_5: $f_1 \in [7,10]$, $f_2 \in [3,10]$, $f_3 \in [5,10]$, $f_4 \in [2,10]$, $f_5 \in [2,10]$, $f_6 \in [1,10]$, $f_7 \in [3,10]$, $f_8 \in [1,10]$, $f_9 \in [1,10]$, Malignant;
R_6: $f_1 \in [5,10]$, $f_2 \in [2,10]$, $f_3 \in [2,10]$, $f_4 \in [4,10]$, $f_5 \in [2,10]$, $f_6 \in [8,10]$, $f_7 \in [1,10]$, $f_8 \in [1,10]$, $f_9 \in [1,10]$, Malignant;
R_7: $f_1 \in [3,10]$, $f_2 \in [3,10]$, $f_3 \in [3,10]$, $f_4 \in [3,10]$, $f_5 \in [5,10]$, $f_6 \in [4,10]$, $f_7 \in [4,10]$, $f_8 \in [1,10]$, $f_9 \in [1,10]$, Malignant;
R_8: $f_1 \in [4,10]$, $f_2 \in [1,10]$, $f_3 \in [1,10]$, $f_4 \in [9,10]$, $f_5 \in [3,10]$, $f_6 \in [4,10]$, $f_7 \in [1,10]$, $f_8 \in [1,10]$, $f_9 \in [1,10]$, Malignant;
R_9: $f_1 \in [7,10]$, $f_2 \in [1,10]$, $f_3 \in [3,10]$, $f_4 \in [1,10]$, $f_5 \in [4,10]$, $f_6 \in [1,10]$, $f_7 \in [4,10]$, $f_8 \in [1,10]$, $f_9 \in [1,10]$, Malignant;
R_{10}: $f_1 \in [4,10]$, $f_2 \in [2,10]$, $f_3 \in [4,10]$, $f_4 \in [4,10]$, $f_5 \in [2,10]$, $f_6 \in [2,10]$, $f_7 \in [3,10]$, $f_8 \in [2,10]$, $f_9 \in [1,10]$, Malignant;
R_{11}: $f_1 \in [5,10]$, $f_2 \in [2,10]$, $f_3 \in [3,10]$, $f_4 \in [1,10]$, $f_5 \in [4,10]$, $f_6 \in [1,10]$, $f_7 \in [2,10]$, $f_8 \in [8,10]$, $f_9 \in [1,10]$, Malignant;
R_{12}: $f_1 \in [9,10]$, $f_2 \in [5,10]$, $f_3 \in [3,10]$, $f_4 \in [1,10]$, $f_5 \in [2,10]$, $f_6 \in [3,10]$, $f_7 \in [1,10]$, $f_8 \in [1,10]$, $f_9 \in [1,10]$, Malignant;
R_{13}: $f_1 \in [6,10]$, $f_2 \in [1,10]$, $f_3 \in [3,10]$, $f_4 \in [1,10]$, $f_5 \in [2,10]$, $f_6 \in [3,10]$, $f_7 \in [2,10]$, $f_8 \in [2,10]$, $f_9 \in [1,10]$, Malignant;
R_{14}: $f_1 \in [5,10]$, $f_2 \in [1,10]$, $f_3 \in [2,10]$, $f_4 \in [1,10]$, $f_5 \in [3,10]$, $f_6 \in [4,10]$, $f_7 \in [4,10]$, $f_8 \in [1,10]$, $f_9 \in [1,10]$, Malignant;
R_{15}: $f_1 \in [9,10]$, $f_2 \in [1,10]$, $f_3 \in [1,10]$, $f_4 \in [1,10]$, $f_5 \in [1,10]$, $f_6 \in [5,10]$, $f_7 \in [1,10]$, $f_8 \in [1,10]$, $f_9 \in [1,10]$, Malignant;
R_{16}: $f_1 \in [1,10]$, $f_2 \in [2,10]$, $f_3 \in [2,10]$, $f_4 \in [1,10]$, $f_5 \in [3,10]$, $f_6 \in [1,10]$, $f_7 \in [3,10]$, $f_8 \in [1,10]$, $f_9 \in [1,10]$, Malignant.

Fig. 5. Extracted rules for the experiments.

TABLE VI
PERFORMANCE ADVANTAGES OF THE EXTRACTED RULES IN FIG. 5

Models	Precision				
	TP *	FN *	FP *	TN *	ACC *
Original Classifier	433	11	7	232	97.36%
Rules in [23]	428	16	10	229	96.19%
Rules in Fig. 5	432	12	9	230	96.93%

* TP, FN, FP, TN, and ACC denote true positive, false negative, false positive, true negative, and accuracy, respectively.

designed from additively HE, which can be implemented by utilizing `libhcs`[3] and `GMP`.[4]

2) *Scheme in [23] (LQNLS20): LQNLS20* is the state-of-art symmetric key encryption-based secure SVM classification scheme, which focuses on the same system model as VSSVMC. We utilize HMAC-256 and AES-CBC-256 to implement *LQNLS20*.

C. Experimental Evaluations

Time Costs of VSSVMC*:* We conduct 1000 experiments to evaluate the total time costs of each algorithm of VSSVMC. As shown in Fig. 6, the total time costs of each algorithm in VSSVMC grow linearly when the number of experimental

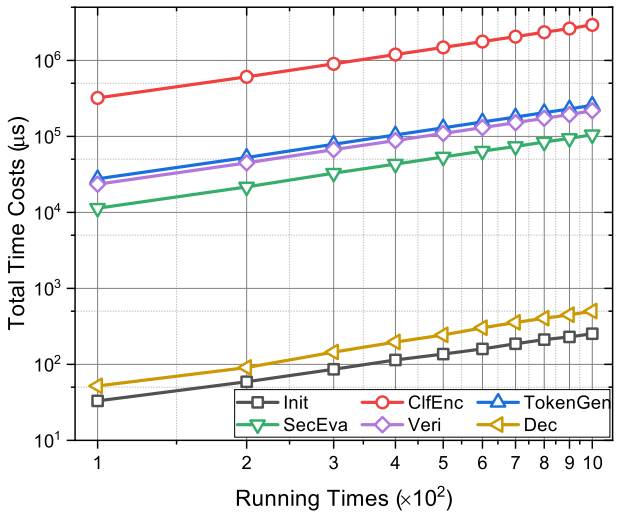

Fig. 6. Total time costs of VSSVMC.

running times grow linearly, which indicate that the time complexity of each algorithm in VSSVMC is constant. Namely, the time complexity of Init, ClfEnc, TokenGen, SecEva, Veri, and Dec are $\mathcal{O}(1)$ when the SVM classifier is pretrained.

Specifically, Fig. 6 depicts that the total time cost of Init is 252 μs when initializing 1000 systems (generating security parameters for 1000 health monitoring systems), which shows that the average initialization time is about 0.252 μs. We test the time costs of ClfEnc by encrypting the extracted rules in

[3]https://github.com/tiehuis/libhcs
[4]https://gmplib.org/

TABLE VII
PERFORMANCE ADVANTAGES OF VSSVMC ON THE BREAST-CANCER-WISCONSIN DATA SET

Schemes	FP [1]	Verifiability	Time Costs in Each Side			Communication Costs		Storage Costs
			$\mathcal{MI}$	$\mathcal{CS}$	$\mathcal{C}$	Tokens	Decisions	Indexes
BPTG15 [16]	N/A	✗	1167μs	N/A	$71026\mu\text{s} + 77572\mu\text{s}^2$	2304 B	256 B	N/A
LQNLS20 [23]	10^{-2}	✗	17654μs	13μs	2097μs	1251 B	137 B	2312 B
LQNLS20 [23]	10^{-4}	✗	26402μs	24μs	3848μs	2151 B	137 B	4292 B
LQNLS20 [23]	10^{-6}	✗	61208μs	50μs	8141μs	4705 B	137 B	5732 B
VSSVMC	N/A	✓	3005μs	107μs	468μs	1725 B	3786 B	47104 B

[1] FP denotes the false positive rate, which is inevitably introduced in Bloom Filter-based schemes, such as **LQNLS20** [23].
[2] We show the time cost of security parameter initialization (one-time) and the time costs of processing cloud-based health monitoring services in the client side (average time costs for each decision request), respectively.

Fig. 5. As shown in Fig. 6, the total time cost of ClfEnc is 2 933 564 μs when encrypting 1000 classifiers, which demonstrates that the average classifier encryption time is about 2934 μs. We also evaluate the time costs of TokenGen by generating encrypted tokens for 1000 random-produced feature vectors. Fig. 6 demonstrates that the total time cost of TokenGen is 256 540 μs, which illustrates that the average token generation time is about 257 μs. For the secure evaluation, verification, and decryption part, we assume that there are three rules matched among the total 16 rules, i.e., $|\boldsymbol{c}(\boldsymbol{v})| = 3$. Fig. 6 shows that the total time costs of SecEva, Veri, and Dec are 105 104, 217 367, and 501 μs when handling 1000 medical decision requests, which demonstrates that the average secure evaluation, verification, and decryption time are 105 μs, 217 μs, and 0.5 μs, respectively. Hence, each algorithm in VSSVMC: 1) achieves $\mathcal{O}(1)$ computational complexity when an SVM classifier is pretrained and 2) requires several microseconds to execute all calculations on the tested data set.

Performance Comparisons With Existing Schemes: We show the performance advantages of VSSVMC in Table VII by comparing the functionality, time costs, communication costs, and storage costs with existing secure SVM classification schemes on the tested data set. Different from the existing schemes, VSSVMC focuses on a malicious threat model and ensures both verifiability and confidentiality in cloud-based health monitoring services. Therefore, VSSVMC is secure against a malicious adversary, while both *BPTG15* and *LQNLS20* only consider the honest-but-curious adversaries.

We evaluate the average time costs of *BPTG15* [16], *LQNLS20* [23], and VSSVMC at medical institute ($\mathcal{MI}$), cloud server ($\mathcal{CS}$), and client ($\mathcal{C}$). Since *LQNLS20* is a Bloom filter-based scheme, the number of hash functions of each Bloom filter in *LQNLS20* is chosen as 5, 8, and 17 when the additional false positive rates of the encrypted indices are $\text{FP} = 10^{-2}$, 10^{-4}, and 10^{-6}, respectively. As shown in Table VII, the time cost at the $\mathcal{CS}$ side is not applicable to *BPTG15* because *BPTG15* focuses on a server–client two-party system model, which inevitably adopts the additively homomorphic technique and leads to a high time costs at the $\mathcal{C}$ side. Therefore, *BPTG15* requires $\mathcal{C}$ to spend 71 026 μs to initialize and generate homomorphic keys and 77 572 μs to encrypt the feature vector and decrypt the encrypted decisions. To avoid the heavy computational costs at the $\mathcal{C}$ side, *LQNLS20* utilizes the Bloom filter technique and outsource both the encrypted classifier and the decision task to $\mathcal{CS}$. With such a design, the time costs of *LQNLS20* at the $\mathcal{C}$ side are reduced to 2097, 3848, 8141 μs when $\text{FP} = 10^{-2}$, 10^{-4}, and 10^{-6}, respectively. While the time costs of *LQNLS20* at the $\mathcal{MI}$ side are 17 654, 26 402, and 61 208 μs when $\text{FP} = 10^{-2}$, 10^{-4}, and 10^{-6}, respectively. However, the adoption of Bloom filter techniques in *LQNLS20* inevitably introduces additional false positive to the encrypted indices, which will further reduce the accuracy of SVM classification. Meanwhile, when choosing a low false positive rates (less than 10^{-2}), the number of hash functions k in *LQNLS20* increases concomitantly, which will further increase the time costs at the $\mathcal{MI}$ side and the $\mathcal{C}$ side. Different from *LQNLS20*, we transform the SVM classifier to an index that is represented as a set of Boolean vectors, which not only avoid the additional false positive to the SVM classifier but also incur a more stable time costs. By utilizing symmetric key encryption, VSSVMC also reduces the time costs at both the $\mathcal{MI}$ side and the $\mathcal{C}$ side. As a result, the time costs of VSSVMC at the $\mathcal{MI}$, $\mathcal{CS}$, and $\mathcal{C}$ sides are 3005, 107, and 468 μs, which demonstrates that VSSVMC improves the computational efficiency in terms of average time costs.

We compare both the communication and the storage costs of VSSVMC with *BPTG15* and *LQNLS20*. We utilize the size of indices to denote the storage costs because both *LQNLS20* and VSSVMC outsource the encrypted classifier to $\mathcal{CS}$. As depicted in Table VII, VSSVMC requires 47-kB storage costs in the tested data set, which is tiny in real-world cloud computing services. Meanwhile, the storage costs of *LQNLS20* are 2.3, 4.3, and 5.7 kB when $\text{FP} = 10^{-2}$, 10^{-4}, and 10^{-6}, respectively. When choosing different false positives, the storage costs of *LQNLS20* are different because the size of indices become larger when the chosen false positive is smaller. Although VSSVMC requires more storage costs than *LQNLS20*, VSSVMC considers a stronger adversary model and ensures the verifiability against a malicious adversary. As a result, the encrypted indices of VSSVMC contain additional pseudorandom strings for decision verification while that of *LQNLS20* does not enable the functionality of decision verification. Note that *BPTG15* focuses on a client–server two-party

system model, which does not need to outsource the encrypted index to $\mathcal{CS}$.

For the communication costs, since $\mathcal{C}$ sends the token to $\mathcal{CS}$ and CS returns the decision to $\mathcal{C}$, we identify both the token size and the decision size as the bandwidth costs of medical decision requests and responses, respectively. As shown in Table VII, the size of tokens and decisions of VSSVMC are 1725 and 3786 B, which are tiny costs in current wireless network throughput. Since the encrypted decisions contain both the ciphertexts of decisions and verification messages, the size of decisions of VSSVMC is larger than that of *BPTG15* and *LQNLS20*. Furthermore, the size of tokens of VSSVMC is similar to that of *BPTG15* and *LQNLS20*.

In summary, VSSVMC: 1) only takes several microseconds to complete the cloud-based health monitoring services with VSSVMC and 2) requires tiny communication costs (less than 6 kB) and storage costs (less than 47 kB) on the tested data set.

VIII. Conclusion

In this article, we have investigated a malicious threat model in cloud-based health monitoring services, and proposed VSSVMC to ensure the verifiability, confidentiality, and efficiency simultaneously. Different from existing secure SVM classification schemes, VSSVMC enables decision verification for detecting $\mathcal{CS}$'s malicious behaviors, such as forging or deleting the decisions. We have given $\mathcal{L}$-security and verifiability definitions, and provided a simulation-based security proof and a game-based verifiability proof for VSSVMC. We have also provided performance analyses to show that VSSVMC achieves $\mathcal{O}(1)$ computational complexity. Furthermore, experimental results based on the Breast-Cancer-Wisconsin data set demonstrated that VSSVMC: 1) costs several microseconds to achieve secure SVM classification and 2) requires little communication costs (less than 6 kB) and storage costs (less than 47 kB). In other words, VSSVMC can perform verifiable and secure SVM classification with acceptable overhead. Therefore, VSSVMC is a potential option to construct trustworthy, secure, and efficient cloud-based health monitoring services. For the future work, we will improve the computational efficiency and reduce the storage costs of VSSVMC in malicious settings.

References

[1] A. S. Abiodun, M. H. Anisi, and M. K. Khan, "Cloud-based wireless body area networks: Managing data for better health care," *IEEE Consum. Electron. Mag.*, vol. 8, no. 3, pp. 55–59, May 2019.

[2] M. W. L. Moreira, J. J. P. C. Rodrigues, K. Saleem, and V. V. Korotaev, "Computational learning approaches for personalized pregnancy care," *IEEE Netw.*, vol. 34, no. 2, pp. 106–111, Mar./Apr. 2020.

[3] 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.

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

[5] M. Li, S. S. M. Chow, S. Hu, Y. Yan, C. Shen, and Q. Wang, "Optimizing privacy-preserving outsourced convolutional neural network predictions," *IEEE Trans. Depend. Secure Comput.*, early access, Oct. 9, 2020, doi: 10.1109/TDSC.2020.3029899.

[6] 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.

[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] 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. Depend. Secure Comput.*, early access, Jun. 14, 2019, doi: 10.1109/TDSC.2019.2922958.

[9] 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.

[10] 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.

[11] L. Zhao, Q. Wang, Q. Zou, Y. Zhang, and Y. Chen, "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1486–1500, 2020.

[12] 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.

[13] C. Zhang, L. Zhu, C. Xu, X. Liu, and K. Sharif, "Reliable and privacy-preserving truth discovery for mobile crowdsensing systems," *IEEE Trans. Depend. Secure Comput.*, early access, May 28, 2019, doi: 10.1109/TDSC.2019.2919517.

[14] Y. Zhang, C. Xu, C. Nan, H. Li, H. Yang, and X. Shen, "Chronos+: An accurate blockchain-based time-stamping scheme for cloud storage," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 216–229, Mar./Apr. 2020.

[15] Y. Rahulamathavan, R. C.-W. Phan, S. Veluru, K. Cumanan, and M. Rajarajan, "Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud," *IEEE Trans. Depend. Secure Comput.*, vol. 11, no. 5, pp. 467–479, Sep./Oct. 2014.

[16] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proc. 22nd Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*, 2015.

[17] J.-C. Bajard, P. Martins, L. Sousa, and V. Zucca, "Improving the efficiency of SVM classification with FHE," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1709–1722, 2019, doi: 10.1109/TIFS.2019.2946097.

[18] H. Zhu, X. Liu, R. Lu, and H. Li, "Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 3, pp. 838–850, May 2017.

[19] O. Ohrimenko *et al.*, "Oblivious multi-party machine learning on trusted processors," in *Proc. 25th USENIX Conf. Security Symp.*, 2016, pp. 619–636.

[20] 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.

[21] J. Liang, Z. Qin, J. Ni, X. Lin, and X. Shen, "Efficient and privacy-preserving outsourced SVM classification in public cloud," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, 2019, pp. 1–6.

[22] H. Yunhong, F. Liang, and H. Guoping "Privacy-preserving SVM classification on vertically partitioned data without secure multi-party computation," in *Proc. IEEE 5th Int. Conf. Natural Comput. (ICNC)*, Tianjian, China, 2009, pp. 543–546.

[23] 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.

[24] Y. Zhang, C. Xu, H. Li, K. Yang, N. Cheng, and X. S. Shen, "Protect: Efficient password-based threshold single-sign-on authentication for mobile users against perpetual leakage," *IEEE Trans. Mobile Comput.*, early access, Feb. 24, 2020, doi: 10.1109/TMC.2020.2975792.

[25] 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.

[26] Y. Liu *et al.*, "Physical layer security assisted computation offloading in intelligently connected vehicle networks," *IEEE Trans. Wireless Commun.*, early access, Jan. 22, 2021, doi: 10.1109/TWC.2021.3051772.

[27] N. Cheng *et al.*, "Big data driven vehicular networks," *IEEE Netw.*, vol. 32, no. 6, pp. 160–167, Nov./Dec. 2018.

[28] F. Lyu *et al.*, "LEAD: Large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics," *IEEE Trans. Mobile Comput.*, early access, Apr. 2, 2020, doi: 10.1109/TMC.2020.2984261.
[29] 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.
[30] 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 35th Int. Conf. Data Eng. (ICDE)*, Macao, China, 2019, pp. 626–637.
[31] J. Zhu, Q. Li, C. Wang, X. Yuan, Q. Wang, and K. Ren, "Enabling generic, verifiable, and secure data search in cloud services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 8, pp. 1721–1735, Aug. 2018.
[32] H. Ren, H. Li, D. Liu, G. Xu, N. Cheng, and X. S. Shen, "Privacy-preserving efficient verifiable deep packet inspection for cloud-assisted middlebox," *IEEE Trans. Cloud Comput.*, early access, Apr. 29, 2020, doi: 10.1109/TCC.2020.2991167.
[33] 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.
[34] S. Lai *et al.*, "Result pattern hiding searchable encryption for conjunctive queries," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2018, pp. 745–762.
[35] X. Fu, C. Ong, S. Keerthi, G. G. Hung, and L. Goh, "Extracting the knowledge embedded in support vector machines," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 1. Budapest, Hungary, 2004, pp. 291–296.
[36] W. Ogata and K. Kurosawa, "Efficient no-dictionary verifiable searchable symmetric encryption," in *Proc. Financ. Cryptogr. Data Security*, 2017, pp. 498–516.

Jinwen Liang (Graduate Student Member, IEEE) received the B.S. degree 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 Broadband Communications Research Lab, 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 of IEEE VTC'19 Fall.

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

He is the Vice Dean and a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, where he serves as the Director of the Hunan Key Laboratory of Big Data Research and Application. 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. He 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 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, the Education Award in 2017 from the IEEE Communications Society, the Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013, the Excellent Graduate Supervision Award in 2006 from the University of Waterloo, and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. 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, a Vice President 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 has served as the Editor-in-Chief of the *IEEE Internet of Things Journal*, *IEEE Network Magazine*, and *IET Communications*. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.