IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 January 2025, accepted 5 February 2025, date of publication 10 February 2025, date of current version 4 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3540261

RESEARCH ARTICLE

Privacy-Preserving Machine Learning (PPML) Inference for Clinically Actionable Models

BARIS BALABAN[1], SEYMA SELCAN MAGARA[2,3], CAGLAR YILGOR[4], ALTUG YUCEKUL[4], IBRAHIM OBEID[5], JAVIER PIZONES[6], FRANK KLEINSTUECK[7], FRANCISCO JAVIER SANCHEZ PEREZ-GRUESO[6], FERRAN PELLISÉ[8], AHMET ALANAY[4], ERKAY SAVAS[2], (Member, IEEE), ÇETIN BAĞCI[9], AND OSMAN UGUR SEZERMAN[1], EUROPEAN SPINE STUDY GROUP

[1]Department of Biostatistics and Bioinformatics, Institute of Health Sciences, Acıbadem Mehmet Ali Aydınlar University, 34638 Istanbul, Türkiye
[2]Department of Computer Science and Engineering, Sabancı University, 34956 Istanbul, Türkiye
[3]Department of Computer Science, University of Tübingen, 72074 Tübingen, Germany
[4]Department of Orthopedics and Traumatology, Acibadem University School of Medicine, 34750 Istanbul, Türkiye
[5]Clinique du Dos, Elsan Jean Villar Private Hospital, 33520 Bordeaux, France
[6]Spine Surgery Unit, Hospital Universitario La Paz, 28046 Madrid, Spain
[7]Spine Center Division, Department of Orthopedics and Neurosurgery, Schulthess Klinik, 8008 Zürich, Switzerland
[8]Spine Surgery Unit, Hospital Universitari Vall d'Hebron, 08035 Barcelona, Spain
[9]Bilmed Computer and Software Company, 34742 Istanbul, Türkiye

Corresponding author: Baris Balaban (Baris.Balaban@live.acibadem.edu.tr)

The work of Erkay Savas was supported by the European Union's Horizon Europe Research and Innovation Program under Grant 101079319.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Acibadem University Medical Research Evaluation Board (Acıbadem Üniversitesi Tıbbi Araştırmalar Değerlendirme Kurulu - ATADEK) under Application No. 2019-11/34, and performed in line with the Declaration of Helsinki.

ABSTRACT Machine learning (ML) refers to algorithms (often *models*) that are learned directly from data, germane to past experience. As algorithms have constantly been evolving with the exponential increase of computing power and vastly generated data, privacy of algorithms as well as of data becomes extremely important due to regulations and IP rights. Therefore, it is vital to address privacy and security concerns of both data and model together with other performance metrics when commercializing machine learning models. Our aim is to show that privacy-preserving machine learning inference methods can safeguard the intellectual property of models and prevent plaintext models from disclosing information about the sensitive data employed in training these ML models. Additionally, these methods protect the confidentiality of model users' sensitive patient data. We accomplish this by performing a security analysis to determine an appropriate query limit for each user, using the European Spine Study Group's (ESSG) adult spinal deformity dataset. We implement a privacy-preserving tree-based machine learning inference and run two security scenarios (scenario A and scenario B) containing four parts with progressively increasing the number of synthetic data points, which are used to enhance the accuracy of the attacker's substitute model. A target model is generated with particular operation site(s) in each scenario, and substitute models are built with nine-time threefold cross-validation using the XGBoost algorithm with the remaining sites' data to assess the security of the target model. First, we create box plots of the test sets' accuracy, sensitivity, precision, and F-score metrics to compare the substitute models' performance with the target model. Second, we compare the gain values of the target and substitute models' features. Third, we provide an in-depth analysis to check the inclusion of target model split points in substitute models with a heatmap. Finally, we compare the outputs of public and privacy-preserving models and report intermediate timing results. The privacy-preserving XGBoost model results are identical to the original plaintext model in the aforementioned two

The associate editor coordinating the review of this manuscript and approving it for publication was Peter Langendoerfer.

scenarios in terms of prediction accuracy. The differences between performance metrics of best-performing substitute models and target models are 0.27, 0.18, 0.25, 0.26 for scenario A, and 0.04, 0, 0.04, and 0.03 for scenario B for accuracy, sensitivity, precision, and F-score, respectively. The differences between target model accuracy and the mean accuracy values of models in each scenario on the substitute models' test dataset are 0.38 for scenario A and 0.14 for scenario B. Based on our findings, we conclude that machine learning models (i.e., our target models) may contribute to the advancement in the field of application where they are deployed. Ensuring the security of both the model and the user data enables the protection of the intellectual property of ML models, preventing the leakage of sensitive information used in training and model users' data.

INDEX TERMS Homomorphic encryption, privacy-preserving machine learning, XGBoost.

I. INTRODUCTION

In today's business world, as well as in academia, it is essential to make data-driven decisions and gain insights using data hoarded in unprecedented amounts, speed and variety. The ever-increasing growth of vastly generated data and breathtaking progress in computing and storage technologies facilitate the development and adoption of superior machine learning techniques with higher accuracy values and faster computation times. Machine learning (ML) techniques, which comprise methods and algorithms that learn from data, have constantly been evolving over the past 25 years, on par with the unrivaled increase in computing power and data collection and storage. The technological breakthroughs and trends in machine learning trends provide companies and organizations incentives to adopt data-driven approaches in modus operandi [1]. As such, machine learning techniques are widely used in various industries, including finance, healthcare, medicine, bioinformatics, and e-commerce. Smartphone usage allows users to benefit from machine learning models by sharing their personal information, such as location and preferences. This sharing of personal information leads to sensitive information leakage from the users if privacy concerns are not properly addressed [2].

Even though machine learning is the backbone of predictive analysis, it is not sufficient to assess them, considering only the functionality and the performance metrics such as accuracy and AUC. It is also vital to address the data and model owners' privacy and security concerns. On par with this argument, Privacy-Preserving Machine Learning (PPML) has become a popular topic in the industry (due to regulations such as GDPR compliance) and the research community [3].

The first solution for privacy concerns is to encrypt all the sensitive information before uploading it to a data store or the cloud. Although offering a safe solution, the encrypted data is useless before decryption. The breakthrough of homomorphic encryption schemes starting with Craig Gently's seminal work that introduced Fully Homomorphic Encryption scheme (FHE), allowed us to compute directly on encrypted data [4] without decryption.

Recently, homomorphic encryption has played a significant facilitator role in the growth of Machine Learning as a Service (MlaaS), as evident by the increasing number of research papers and practical applications, utilizing HE for privacy-preserving purposes in ML [5]. The users of machine learning models benefit from privacy-preserving computation, which allows them to put their sensitive data into good use, free from security concerns. On the other hand, the models counted as intellectual property, and storing it in a trusted server protects the IP and business interest thereof [2].

When the privacy of the input data is assured, more and more parties started to use such models, which results in continual improvement in homomorphic encryption schemes and machine learning models. Amazon Web Services (AWS) is one of the most comprehensive cloud platforms, which also implements homomorphic encryption schemes that allow model training and inference on encrypted data [6].

Privacy-preserving machine learning (PPML) has become a key focus in healthcare, leading to a diverse range of approaches that protect sensitive patient data at various stages. Some works rely on multi-party computation such as Parra-Ullauri et al. [7] , demonstrating moderate slowdowns for encrypted logistic regression. Others, such Lee et al. [8] and Chen et al. [9], apply homomorphic encryption to boosting or deep models, achieving promising privacy gains but typically centering on training-phase confidentiality. Meanwhile, studies leveraging differential privacy, including Wei et al. [10] , enable federated training across multiple institutions at the cost of some model accuracy. Finally, Wu et al. [11] and Hassan et al. [12] have shown how decision-tree-based methods can be encrypted for healthcare classification tasks, though these solutions often concentrate on data protection alone rather than robustly guarding the model's intellectual property.

As MLaaS increases in popularity, more organizations target using their ML assets without disclosing the models themselves. European Spine Study Group (ESSG) is an example, which was founded in 2010 by European spinal deformity surgeons to develop an adult spinal deformity database to evaluate clinical outcomes and mechanical complications. Tiwari et al. [3] suggested that there is still room for research on inference phase rather than other machine learning steps in the near future, as most of the past studies focus on data aggregation and training phases. In this study, we aim to implement a privacy-preserving inference scheme based on homomorphic encryption for the models

generated with the ESSG data and perform a security analysis of the model stored in a trusted server, as the MLaaS give out information about the model in every query.

Our motivation is to show that homomorphic encryption schemes can be employed in privacy-preserving machine learning inference to protect both the privacy of the ML models and sensitive user data. One of the primary challenges in PPML is maintaining the accuracy and performance of ML models while implementing encryption and other privacy-preserving techniques. Often, these measures can dilute the model's effectiveness, leading to a trade-off between privacy and performance. Secondly, maintaining the ML models' intellectual property (IP) poses a significant challenge in our study. To protect IP on the models developed, we provide a security analysis to define an appropriate query limit per user with ESSG's adult spinal deformity dataset, as unlimited access to query models may result in constructing as accurate substitute models from the query results as the target model. Healthcare data security encompasses a wide range of threats, from vulnerabilities in biomedical IoT devices [13] to the complexities of secure enclaves [14] . By focusing on encrypted inference, our system partially avoids pitfalls associated with insecure edge nodes or untrusted hardware. Incorporating these broader security considerations complements the present study's focus on inference confidentiality.

A. OUR CONTRIBUTIONS

In this section, we outline our contributions to the field and list them under the three main categories.

- A novel privacy-preserving prediction method: This study adapts our novel approach introduced in [15] to perform machine learning inference on encrypted medical data.The method employs specialized encoding techniques for both the model and the data, enabling the XGBoost algorithm—or any tree-based machine learning algorithm—to operate without decrypting the data, thereby maintaining the confidentiality of sensitive information. By adopting an efficient encoding approach akin to one-hot encoding and utilizing logical gate operations for comparisons, the study successfully mitigates the computational challenges typically linked to comparison operations on encrypted data. This approach guarantees that the model can execute the required computations both swiftly and precisely.
- Comprehensive security analysis: In this study, we conduct a thorough security analysis to evaluate the robustness of the privacy-preserving methods against potential attacks. In our analysis, we use realistic adversarial models, where an attacker can work alone or colluding with others to compromise the machine learning model. We also show that the patient data can be secured with cryptographic techniques.This analysis is essential for identifying vulnerabilities in PPML models and ensuring their robust defense against malicious activities without compromising the intellectual property of the inference algorithms. As detailed in the related work Section II-C1, the XGBoost algorithm has been demonstrated to be effective in predicting mechanical complications in ASD patients. Consequently, in our security scenarios, we employed the XGBoost algorithm for both the target and substitute models to leverage its predictive capabilities. Our analysis suggests that protecting the inference model is not possible in realistic scenarios where the model is relatively simple due to explainability requirements and care must be taken to protect the output of the model such as limiting the number of queries. We also show a methodology to determine a threshold for the number of allowed queries to keep the model private.
- Implementation and a practical application in a real-world scenario: We provide a practical implementation of our approach using an open-source homomorphic encryption library, which demonstrates the feasibility of deploying advanced cryptographic techniques in real-world settings. This aspect of the study illustrates the scalability and adaptability of the proposed methods, affirming their applicability across a broad spectrum of applications.

II. BACKGROUND

In this section, we provide the background knowledge relevant to the discussions in the subsequent sections.

A. HOMOMORPHIC ENCRYPTION

Homomorphic encryption enables calculations on ciphertexts, where results are the same as the plaintext operation when we decrypt the resulting ciphertext. Homomorphism in algebra is a map between two mathematical objects while preserving their structures. To define homomorphic encryption, there should be a mapping from the algebraic object A to another one B, $f : A \rightarrow B$, such that we have $f(a) \times f(b) = f(a \otimes b)$ holds for operations $\times$ in A and $\otimes$ in B. The encryption is additively homomorphic if $f(a + b) = f(a) \boxplus f(b)$ and multiplicatively homomorphic if $f(a \cdot b) = f(a) \odot f(b)$ [2], [4]. An encryption is partially homomorphic if it only allows one operation, such as RSA and ElGamal schemes. Gentry et al. proposed a fully homomorphic encryption (FHE) scheme in 2009, in a seminal work [4], where they showed that it is possible to support both operations of algebraic structure using *noise* to encrypt messages. The authors proposed an ingenious idea captured in *bootstrapping* operation, which allows theoretically unlimited number of homomorphic operations in the presence of ever-increasing noise in the ciphertext after each homomorphic operation, which renders the ciphertext undecryptable after a certain number of such operations. The multiplication operation, in particular, increases the noise level significantly. The solution is to apply bootstrapping just before the noise budget is exhausted, which decreases the noise level by generating a fresh ciphertext with a new noise budget. Although significant progress is achieved each year

in developing better implementation of FHE schemes [16], [17], in this work, we use somewhat homomorphic encryption schemes, where expensive bootstrapping is avoided, and only available noise budget is used.

Comparison of HE with other PETs: In our work, we assume the existence of an XGBoost model hosted by a server where clients send their queries to obtain predictions. There are different cryptographic techniques to provide data privacy, including homomorphic encryption (HE), secure multi-party computation (MPC), and differential privacy (DP). Both homomorphic encryption (HE) and secure multi-party computation (MPC) are strong candidates for protecting the client's data during inference, while DP is less suited for this setting. DP introduces noise to ensure privacy, which creates a trade-off between privacy and accuracy. However, it does not fully protect the client's data during computation, making it unsuitable for scenarios where exact privacy is essential. Conversely, MPC provides complete privacy for client data through distributed computation. Despite its stronger privacy guarantees, MPC suffers from significant communication overhead, making it impractical for large-scale applications. In contrast, HE allows computations directly on encrypted data, ensuring robust privacy while avoiding MPC communication bottlenecks. Adopting HE for complex tasks might be challenging due to its computational limitations, particularly for non-linear functions and comparison operations. However, our method addresses these challenges by simplifying costly comparison operations into more efficient multiplications. Multiplication operation leverages batching and NTT techniques in modern HE schemes, significantly improving performance. As a result, We have chosen HE as the main privacy-enhancing technology of this work, and our framework demonstrates that HE is now a practical and efficient solution for privacy-preserving XGBoost inference, overcoming its traditional limitations

Post-Quantum Security and Efficiency Enhancements: Our work aligns closely with lattice-based cryptographic techniques, which form the backbone of many post-quantum cryptographic (PQC) schemes. For instance, using key encapsulation mechanisms (KEMs) mirrors the security assumptions of lattice-based key exchange (KE) schemes. Both approaches rely on the hardness of problems like Learning With Errors (LWE) or Ring-LWE, ensuring resilience against quantum attacks. As a result, our proposed method inherits strong post-quantum security guarantees.

Moreover, the efficiency of our homomorphic encryption-based approach is further enhanced by the integration of the Number Theoretic Transform (NTT), a key optimization in lattice-based cryptography. The use of NTT significantly accelerates polynomial arithmetic operations, which are foundational to both KEMs and similar PQC schemes. Thus, if NTT computations are optimized, schemes akin to ours can achieve both high security and impressive performance, reinforcing their practicality for post-quantum applications.

A class of practical FHE schemes allow concurrent execution of the same homomorphic operation on multiple data thanks to the *batching* technique, where many data points can be encoded into the *slots* of one ciphertext. This way, with one homomorphic operation, many data points are homomorphically evaluated.

B. XGBoost ALGORITHM

XGBoost is a supervised decision tree ensemble method inspired by the gradient boosting method [18], which uses gradient descent algorithms to boost weak learners. Algorithms are additive models, meaning the algorithm builds a new weak classifier with the previous model's residuals at every iteration. Hence, every weak classifier built must consider what the previous models fail to capture since the algorithm does not allow any change in the previously made classifiers. Bagging (bootstrap aggregation) is another technique in the XGBoost algorithm to increase the model's accuracy by selecting a ratio of the input data each time a tree is built. Boosting allows the constructed model to be more robust as it will reduce the variance of a data set with noise common in real-world applications [19].

C. RELATED WORK

In this section, we provide preliminary background on two papers on which the current work is based.

1) BUILDING CLINICALLY ACTIONABLE MODELS WITH XGBoost ALGORITHM

Our work can be considered as the continuation of the work ''Building Clinically Actionable Models for Predicting Mechanical Complications in Postoperatively Well-Aligned Adult Spinal Deformity Patients using XGBoost Algorithm'' [20]. In the paper, a subgroup of adult spinal deformity (ASD) patients is analyzed, namely the proportioned subgroup. After correlation elimination, nine times threefold cross-validation is used for hyperparameter optimization. The best parameter set performance metric is found to be the F-score. The scale positive weight parameter is defined by negative instances over positive instances, and the subsample and column sample parameters are set to the default (1). A final model is built with all training data after cross-validation. To reduce the complexity of the model, the bagging technique is not used. Therefore, the best feasible parameter set becomes less complex.

The feature selection process is continued in the model-building phase to eliminate non-sensible and non-explainable features. With the hyperparameter selection method explained in the above paragraph and the expert-guided feature elimination in consecutive model-building stages, clinically explainable models are obtained. The expert-guided model produces four trees and 16 leaves with seven features. The model achieves 74% accuracy, 80% sensitivity, 73% specificity, and 95% negative predictive

value, while the precision is 36%. To summarize, the proposed model to predict mechanical complications in [20] is a calibration of XGBoost algorithm aiming to produce clinically actionable models.

2) PRIVACY PRESERVING MACHINE LEARNING INFERENCES FOR TREE-BASED ALGORITHMS

There are two common approaches in privacy-preserving machine learning studies: differential privacy and homomorphic encryption. Differential privacy safeguards a model by preventing unauthorized access to training data, while homomorphic encryption uses a server as an entirely opaque intermediary between the data holders, delivering computational power without accessing any plaintext data. However, these approaches vary greatly and come with their own specific challenges: differential privacy typically requires adding perpetual and limitless noise, whereas homomorphic encryption is limited to working with numerically encoded values that have a strict bit-length limitation [21]. Although the use of differential privacy in PPML delivered promising results with a minimum reported F-score 0.85 in [22], 0.94 in [23] for naive Bayes classification and 0.88 in [24] where the authors tried commonly used machine learning algorithms with five different health datasets, we applied homomorphic encryption in our study as our project requirements align with the usage of HE.

Our work can also be identified as a real-world application of the methodology proposed in the paper ''ML with HE: Privacy-Preserving Machine Learning Inferences for Genome Studies'' [25]. The proposed method eliminates the need for comparison operations in the inference phase of privacy-preserving machine learning by leaking some non-essential information about the model. For example, in [25], the split points used in the target model alongside with random ones are disclosed to model users for them to construct their encrypted query. While privacy indications of leaking such information for practical usage of the model are discussed in the study, disclosing split points is found to be necessary for efficiency and effectiveness in practicable use of models [26], [27]. Section IV provides more information about their work.

Before [25] two methodologies were proposed comparison function of HE algorithms [28], [29], and those methods were inefficient in practice [30] as they need many iterations in their algorithms which causes high level of noise and longer execution times. By following [25] and its encoding algorithm, we propose an efficient inference methodology for mechanical complication prediction in ASD patients by expressing comparison operations with multiplication operations and disclosing insensitive information about the model, i.e split points of the model with the arbitrarily selected random ones, the encoding rules. This inference algorithm was tried with various public datasets and the results were reported and compared with the literature in [15] and [25].

III. METHODS AND DATA SETS

This section gives detailed information about the data set and the methods used in our study.

A. THE ESSG DATASET

Medical data, which is essential for discovering disease trends, enabling personalized treatments, and advancing healthcare technology, is protected by HIPAA in the United States and GDPR in the European Union to ensure patient privacy rights. Therefore, the security and privacy of medical datasets must be maintained at all times. As ESSG ensures the confidentiality of its adult spinal deformity data set, the authors of this study do not have the option to publish patient records. Below, the ethics approval and consent to participate and the inclusion criteria of the study are explained briefly.

Institutional review board approval was obtained for patient enrollment and data collection protocols at each site. This study was approved by the Acıbadem University Medical Research Evaluation Board (Acıbadem Üniversitesi Tıbbi Araştırmalar Değerlendirme Kurulu – ATADEK) with the approval reference number 2019-11/34. This study took place in Acıbadem University and ATADEK is the medical research evaluation board of Acıbadem University. Each patient provided written informed consent before participation in the study. All the methods were carried out following the Declaration of Helsinki.

The ESSG dataset used for the analysis consists of data from 6 European sites and 244 patients collected between 2010 and 2019. Patients included in the study are at least 18 years old and have at least one of the following measurements: Cobb angle $\leq$ 20°, sagittal vertical axis $\leq$ 5 cm, pelvic tilt $\leq$ 25°, or thoracic kyphosis $\leq$ 60°. Furthermore, patients underwent at least four levels of posterior instrumented fusion, had a postoperative GAP Score of 0, 1, or 2 (postoperatively well-aligned), and had at least two years of follow-up. In total, 196 features are included in this study, consisting of patients' characteristics, radiographic measurements, technical details, and patient-reported outcome measures. The response variable of the dataset is *mechanical complications*, a combination of the various mechanical complications seen after ASD surgeries [31]

Besides age, body mass index (BMI), follow-up time after surgery, and the estimated blood loss in operation, there are also more technical features used in this study; namely, LIV location, the number of levels fused multiplied by construct location, and pre-operative ODI Walking score. LIV location is an ordinal feature to locate the lowest instrumented vertebrae. Spinal fusion is an operation that permanently joins together one or more bony vertebrae of the spine to eliminate motion. For instance, a two-level fusion fuses three vertebrae together, while a three-level fusion joins four vertebrae. Next, construct location is the feature that shows which parts of the spine are operated, and the encoding starts from 0 and increments by one in the given order: Thoracic only vs lumbar only vs thoraco-lumbar vs lumbo-sacral

vs thoraco-lumbo-sacral vs cervico-thoracic vs cervico-thoracolumbar vs cervico-thoraco-lumbo-sacral. Also, Oswestry Disability Index (ODI) is the most common patient-recorded outcome measure related to pain. For more information on the dataset, one can profitably refer to [20].

In the security analysis section of the work, we split the dataset into target and substitute model sets and assume that the latter is under the control of the adversary. We reserve test sets using stratified sampling for both groups. We name the training/test data sets of the target model as the *training/test sets* and training/test data sets of the substitute model as the *attacker training/test sets*. The response variable of the attacker training data is set to target model predictions, and in all other datasets, the actual mechanical complication outcome is used.

B. PRIVACY-PRESERVING XGBoost INFERENCE

An XGBoost model is an ensemble of boosted trees where the leaf nodes carry scores. The inference procedure begins by passing and evaluating the input features through each tree, where the model calculates the final prediction score by summing up the individual tree scores. For this purpose, the input features should be compared to the split points of the nodes determined in the training phase. However, the comparison operations performed during prediction can reveal sensitive information about the input data or the underlying model. One practical approach to ensuring data privacy is using homomorphic encryption (HE) in XGBoost. HE allows performing computations on encrypted data without disclosing sensitive information. Nevertheless, certain operations, such as comparison, are computationally demanding over encrypted data with the HE. To alleviate the cost of the comparison operation, Mağara et al. [25] proposed an efficient encoding method that formulates the comparison operation in terms of multiplication operations. In this study, we adopt the proposed method, whose details are provided in Section IV, to protect the developed XGBoost model against model stealing attacks. Also, we provide a security analysis of the method and possible leakage scenarios in Section V-D.

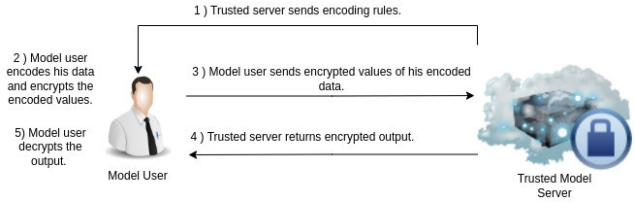


FIGURE 1. Client-server model and the flow of the privacy-preserving XGBoost inference algorithm.

For the implementation of our method shown in Figure 1, we utilize the open-source homomorphic encryption library, PALISADE (currently known as OpenFHE[1]). The library provides support for various HE schemes, including BFV and HEAAN (a.k.a. CKKS). The BFV scheme is well-suited for integer computations, while the CKKS scheme supports real numbers and approximate computing [2], [26]. As we work with real numbers, we implement our XGBoost inference algorithm using the CKKS scheme of the PALISADE library following the work employed in [25]. The CKKS scheme has IND-CPA level security, and it is based on the hardness assumption of the ring learning with errors (RLWE) problem [29].

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

C. TARGET MODEL CROSS-VALIDATION AND PERFORMANCE METRICS

We train our target models in security scenarios to predict mechanical complications in proportionally aligned ASD patients according to the Global Alignment and Proportion (GAP) score methodology [31]. Combining all the mechanical complications as performed in [31], we can formulate a binary classification problem, where we define the presence of mechanical complication(s) as positive instances and healthy patients as negative instances. However, the GAP methodology is insufficient to predict the mechanical complications itself, and other pre-operative, operative, and post-operative features help increase the predicting power [32], [33], [34]. Therefore, we use the features found significant by the clinically explainable model of the work in [20]. These are LIV (lowest instrumented vertebra) location, estimated blood loss, pre-operative ODI walking score, age, BMI, the number of levels fused multiplied by construct location, and follow-up time after surgery.

Since it is crucial to predict possible mechanical complications because there is less possibility of suffering from a mechanical complication in the proportionally aligned subgroup of ASD patients, we set the scale positive weight parameter to the ratio of negative instances over positives to give more weight to positive cases [35]. We use the default parameters (i.e., 1) for column and row sample parameters. The close hyperparameter space is searched to find the best hyperparameter set using F-score as the performance metric. Nine times threefold cross-validation is used for hyperparameter optimization. The final models are built with all training sets. With our target model training methodology, we are able to produce target models that are similar to the clinically explainable XGBoost model in [20].

The means (and standard deviation) of the area under the receiving operating curve (AUROC), sensitivity, specificity, area under the precision-recall curve (AUPRC), precision, and F-score are calculated for cross-validation analysis. For the target models, confusion matrices and accompanying metrics of the training, target, and attacker test sets are calculated. Even though we present AUC and AUPRC values in the cross-validation analysis to compare the models, 0.5 is used as the default threshold value to predict mechanical complications when we build the target models. That's why the tables of cross-validation and the final target and substitute models have accuracy results instead of AUC and

AUPRC values. Also, the feature importance graphs are provided using gain values.

IV. IMPLEMENTATION OF PRIVACY PRESERVING XGBoost INFERENCE

The XGBoost inference algorithm requires comparison operations at each node of the model trees. Due to its prohibitive cost, the homomorphic comparison operation is the most significant barrier to XGBoost inference working on encrypted data. Therefore we use the encoding method for privacy-preserving XGBoost inference following the solution by Magara et al. [15]. The methods used to encode data and the model are based on the split points in the XGBoost trees. The encoding approach relies on the split points of the XGBoost trees to encode the data and the model. By encoding the data based on its value range, we can represent the comparison result as either 1 or 0. This is achieved through digit-by-digit multiplication of the data and model encodings. Consequently, the complex homomorphic comparisons are transformed into simpler homomorphic multiplications, as described below. The explanations of terms used in this section are presented in Table 1.

TABLE 1. Explanation of terms.

Term	Explanation
s	The split point used in the decision rules encoding algorithm.
B	A set of intervals or buckets used to encode split points and feature values.
E_s	The encoded binary vector representing the position of the split point s relative to the predefined intervals in B.
f	A specific feature value from the query data that needs to be encoded using the buckets B.
E_f	The encoded binary vector for the feature value f .
y	Encoded binary values of the node split points.
z	The final boolean expression that represents the result of a homomorphic comparison operation.
z_i	The comparison result of node i in the decision tree used in the tree score calculation algorithm.
c_i	Scores associated with the leaves in a decision tree.
p_1, p_2, p_3, p_4	Algebraic expressions representing the values of the paths in a complete binary tree with a depth of 2.
C_f	Ciphertexts containing the encoded feature values of the client data.
F_j	The ciphertext of feature values organized with the same order as the jth nodes of the model.
M_j	Represents the model's jth node in the homomorphic comparison operation.
Z_j	The ciphertext containing the comparison results of the jth nodes of all trees.
d_{max}	Depth of the trees in an ensemble
n_{TT}	The total number of trees in the model
n_E	The encoding length

The training stage of the XGBoost algorithm determines the optimal splitting points and features, resulting in various split point values for a feature. Magara et al.'s method leverages these split points to encode both the model and the query data. To further obscure the model and prevent clients from gaining insights into the underlying structure, the model owner can add more random split values as encoding rules instead of using the original split points only. This addition ensures that the split values clients receive are not purely derived from the data and the model, adding a layer of randomness and making it harder for clients to reverse-engineer or infer the details of the model. Below, rules for encoding the decision rules and the client data are explained, and pseudocodes can be found below, titled Algorithm 1 and 2.

A. ENCODING OF THE DECISION RULES

The decision rules encoding algorithm encodes a given split point s using a series of buckets represented by B. This encoding procedure is used to create a binary representation, E_s, based on the position of s relative to predefined intervals (buckets). Each bucket, B_i, is defined by a range $[s_i, s_i + 1)$, indicating that it includes s_i but does not include $s_i + 1$.

The encoding starts by initializing E_s as a zero vector, with the length equal to the number of features n_F. The algorithm then iterates over each bucket B_i in the set B. For each bucket, it checks if the split point s is greater than or equal to the lower boundary s_i of the bucket. If s falls within this criterion, the corresponding index i in the vector E_s is set to 1, signifying that s is at least as large as s_i.

The resulting vector E_s is a sparse binary vector where each element corresponds to a bucket and is set to 1 if the split point s meets the criterion of that bucket. The process effectively translates the continuous value s into a discrete binary format that simplifies further modeling or decision processes. The encoded vector E_s is then returned as the output of this procedure, providing a structured way to represent where the split point s falls relative to a series of defined intervals.

Algorithm 1 Decision Rule Encoding

Input Split point of the rules s and the buckets B
Output E_s: The encoding of value s

```
procedure EncodeRule(s, B)
    E_s ← ⟨0, 0, ..., 0⟩                    ▷ |E_s| = n_F
    for each bucket B_i in B do
        if s ≥ s_i then                      ▷ B_i = [s_i, s_{i+1})
            E_s[i] ← 1
        end if
    end for
end procedure
```

B. ENCODING OF THE CLIENT DATA

The client data encoding algorithm is designed to encode a specific feature value, f, from a query using a predefined set of buckets, B. This encoding process translates the continuous or categorical feature value into a binary vector format. The encoding begins by initializing a zero vector E_f, with the length equivalent to the number of buckets. Each bucket B_i in the set B defines an interval, represented as $[s_i, s_i + 1)$, indicating a range that includes s_i but excludes $s_i + 1$.

The procedure iterates over each bucket B_i. For each bucket, the algorithm checks if the feature value f falls within the bounds of B_i. If f is found to be within a bucket, the corresponding index i in the vector E_f is set to 1. This

indicates that the feature value belongs to the interval defined by that particular bucket. Once f is assigned to a bucket, the loop breaks, ensuring that E_f remains a sparse vector where only one element is set to 1, corresponding to the bucket containing the value f. The vector E_f serves as the encoded representation of the feature value f.

Algorithm 2 Query Data Encoding

Input Feature value f and the buckets B
Output E_f: The encoding of the feature F

```
procedure EncodeQueryData(f, B)
    E_f ← ⟨0, 0, ..., 0⟩                ▷ where |E_f| = |B|
    for each bucket B_i in B do
        if f ∈ B_i then
            E_f[i] ← 1
            break
        end if
    end for
end procedure
```

C. COMPARISON USING THE ENCODINGS

For example, let E_f be the encoding of a feature f, and E_s be the encoding of a split point s. Then, their homomorphic multiplication results in all zeros if E_f is not smaller than E_s. Otherwise, it contains a single digit which is one. The following example shows two sample comparison operations using the adopted encoding methods:

$$\begin{array}{rrr} 150: & 0\,0\,1\,0\,0 & \\ 147: & \underline{\circ\ 1\,1\,0\,0\,0} & \\ 150 \overset{?}{<} 147: & 0\,0\,0\,0\,0 & \end{array} \qquad \begin{array}{rrr} 150: & 0\,0\,1\,0\,0 & \\ 153: & \underline{\circ\ 1\,1\,1\,0\,0} & \\ 150 \overset{?}{<} 153: & 0\,0\,1\,0\,0 & \end{array}$$

Here, the first and the second operands represent the data and split point values, respectively. Note also that the number of split points is limited to 5 in the above example and 16 in our study, and for the data, one-hot encoding is used.

D. CALCULATING TREE SCORES

Using this encoding, we can effectively compare an encrypted feature's value with a node's value within the model with a single homomorphic multiplication. If the feature value is smaller, the result is encoded as 1; otherwise, it is 0. It's important to note that the prediction data remains encrypted throughout this process. Consequently, the comparison result is also encrypted, allowing us to navigate to the correct leaf node without the need for decryption. The boolean nature of the node values facilitates this.

In the structure of complete binary trees with a depth of 2, there are precisely four possible paths as shown in Figure 2. The comparison value of node i is indicated with z_i and the scores associated with the leaves are denoted by c_i. From this configuration, the values of the paths are expressed algebraically, which allows for determining the outcome path without decrypting the values, thus maintaining the encryption throughout the process.

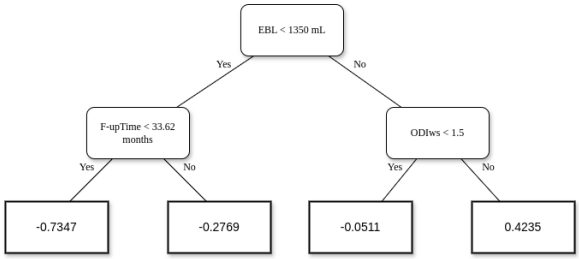


FIGURE 2. Target model's first tree in the initial security scenario.

Using the structure of complete binary trees with a depth of 2, the formulae for calculating the paths are defined as follows:

$$\begin{aligned} p_1 &= z_1 * z_2 * c_1 \\ p_2 &= z_1 * (1 - z_2) * c_2 \\ p_3 &= (1 - z_1) * z_3 * c_3 \\ p_4 &= (1 - z_1) * (1 - z_3) * c_4 \end{aligned}$$

In this configuration, only one path, the correct one, will have a nonzero value at any given time, while the values for the other paths will be zero. Thus, the summation of all p values results in the score of a single tree. This leads to a simplified expression for a single tree's score where l's represent the leaf node scores:

$$score = (z_1 - 1) * l_3 * z_3 + (z_2 * l_1 + l_2) * z_1 + l_4$$

The Tree Score Calculation algorithm determines the cumulative score based on internal node values and leaf values in a binary tree. Each internal node in the set Z contains comparison results, while the leaf values are contained in set L. The algorithm processes each path from a leaf to the tree's root, dynamically building a path list by evaluating whether to take a node's direct value or its complement, depending on the node's index. This decision simulates the traversal through either the left or the right child of a binary tree node. For each path, values are combined using a function, BinaryTreeMultiplication, which presumably multiplies all the values along the path. Each resultant path value is then multiplied by its corresponding leaf value to determine that path's contribution to the overall tree score. Finally, the contributions from all paths are summed to compute the TreeScore, which is the algorithm's output.

The comparison operations should be performed at each internal node for the XGBoost inference. The batching feature of homomorphic encryption enables homomorphic evaluations of comparisons on many multiple nodes simultaneously. Hence, the encodings of all model nodes with the same index are encrypted into the slots of a single ciphertext for computation efficiency. Below we explain how the batching feature is utilized to implement the XGBoost inference using this encoding method.

E. XGBoost INFERENCE PROTOCOL

The client initiates the inference protocol by connecting to the server. First, the server sends the encoding rules, including the split points of the features and the batch size. The batch size is the length of the vector to be encrypted, which depends on the number of trees. It is suggested to use a higher value instead of the required length to hide the positive correlation between the input array lengths and the number of trees. In line with this, we select the maximum batch size permitted in our CKKS setting. Furthermore, the model owner adds some random points to the split point set to conceal the actual split points. These modifications affect neither the model's performance nor accuracy due to the batching feature. We fixed the batch size at 4,096, corresponding to half of the minimum required ring dimension (8,192) in our CKKS setting. Also, the number of split points is fixed to 16 for each feature by adding random values.

In addition, the protocol hides the order of the features in the nodes. To this end, the client is required to encode the feature values and create a ciphertext C_f for each feature separately by concatenating their encoding value.

$$C_f = Enc(E_f||E_f||\dots||E_f).$$

Then, the features are ordered privately at the server side using masks ($Mask$) produced by the model owner as shown in Equation 1.

$$F_j = \sum_{i=0}^{m-1} Mask_j^i \circ C_{f_i}. \tag{1}$$

In Eq. 1, F_j refers to the ciphertext of feature values of the client data, which are organized with the same order as the j^{th} nodes of the model, and m is the number of features.

After the ordering phase, the comparison operation for j^{th} nodes of all trees is applied by homomorphically multiplying F_j with the model ciphertext for j^{th} nodes. This operation is explicitly formulated in Equ 2

$$Z_j = M_j \circ F_j, \tag{2}$$

where M_j is the j^{th} node of the model and Z_j is the ciphertext containing the comparison results of the j^{th} nodes of all trees.

To calculate tree scores, the comparison results of nodes on a path from the root node to a leaf node are homomorphically multiplied, which will be the score of the corresponding leaf node. If all comparison results on a path are non-zero, the score of the corresponding leaf node will be the score of its tree, as the scores of other leaf nodes on the same tree will be 0. Then, the scores of the leaf nodes of the same tree are summed to compute the tree's score. Since the selected leaf node is not revealed, this method does not leak information about the query data to the server. The explanation of the tree score calculation algorithm is given previously and the pseudocode can be found in [15] on page 27.

Finally, all tree scores are added using homomorphic operations such as rotation and summation. After decryption, values smaller than 0 correspond to minimal/no risk of mechanical complications. This number is homomorphically multiplied with a random positive number, and the rest is multiplied with 0 to hide the tree score information before the decryption. This part is referred to as final masking.

The multiplicative depth and scaling factor are two other important parameters in the CKKS scheme. Multiplicative depth is the number of consecutive multiplication operations in the algorithm. Additionally, homomorphic operations, especially multiplication operations, add up noise in the ciphertext. To minimize the noise/error growth in the ciphertext, CKKS schemes scale up the value by **scaling factor** bits, making the scaling factor an indicator of precision in the CKKS scheme. Following [25], we set the parameters multiplicative depth and scaling factor bits of the CKKS object parameters to 5 and 16, respectively. For more detail, readers are referred to the original work [15].

Model information in our proposed model consists of four files. The first step of the model owner is to produce 16 random split points for all features including the target model split points which is the first file. For example, the encoded values of the first tree's nodes shown in Figure 2 are 6, 4, and 2 for EBL, F-upTime, and ODIws nodes, respectively. The second file is the split information file. It is a binary file of 288 bits considering three nodes of six trees and 16 possible split points in scenario A. There are three nodes since both of our target models in the security scenarios have depth two. The third file is the masking file for each feature which is 56,488 bits considering six trees, 16 possible split points, three nodes, and 196 features. The masking file has 1 for consecutive 16 bits if that feature is used in that specific node. Lastly, leaf scores are stored in the trusted server by multiplying them by ten thousand to remove the fractions from the scores. A model user queries the model by sending each of the encoded features. Even though the batch size is selected as 4,096 bits to hide the tree counts of the target models, only the first 96 bits are used to produce the output for each feature for a six-tree-long ensemble.

F. COMPUTATIONAL COMPLEXITY

A complete analysis of the utilized inference method is given in [15], and they discuss the effect of the model complexity with the support of the experimental results in Sections 4.4.1. and 4.4.2. We also provided the tree score calculation method earlier, along with explanations of the node comparison and inference steps. Based on these and the complete analysis given in [15], the multiplication depth of the circuit for the homomorphic inference operation mainly depends on the model's depth, d_{max}. According to the tree score calculation method, computing tree scores requires $\lceil \log_2 d_{max} \rceil$ multiplications. Additionally, masking and ordering, node comparison, and final masking each require one multiplication. Consequently, the total multiplication depth of the circuit is $\lceil \log_2 d_{max} + 4 \rceil$. In our method, where the tree depth is 2, the multiplication depth is $\lceil \log_2 2 + 4 \rceil = 5$.

Moreover, another computationally intensive operation in homomorphic encryption is the rotation function. The

number of rotation operations required is determined by the model complexity, which depends on the total number of trees in the model (n_{TT}) and the encoding length (n_E). The relationship between the rotation operation and the model complexity is also logarithmic, resulting in $\log_2 n_E + \log_2 n_{TT}$ rotation operations necessary for the entire inference protocol.

In conclusion, the computational complexity of our homomorphic method is dependent on the complexity of the XGBoost model. As the model becomes more complex, characterized by an increase in the number of trees, the number of split points, and the depth of the trees, the computational complexity of our solution also increases. However, this increase is not linear; it scales logarithmically with these factors, making the change sublinear. This logarithmic change means that the computational demands grow with sublinear to model complexity specifically, in proportion to the logarithm base 2 of the number of trees, split points, and tree depth.

V. SECURITY ANALYSIS

A. SECURITY ANALYSIS OF HOMOMORPHIC ENCRYPTION

Homomorphic encryption (HE) has emerged as a foundational building block for secure computation, offering robust protection against classical and quantum adversaries. As a post-quantum cryptographic (PQC) secure method, HE relies on the hardness of lattice-based problems such as Learning With Errors (LWE) or its variants, which resist attacks from conventional and quantum computers. This resilience ensures that HE provides a strong basis for protecting data confidentiality in sensitive applications, such as privacy-preserving machine learning. While the theoretical security of HE is well-established, its practical implementations may still be vulnerable to various types of attacks that exploit weaknesses beyond the cryptographic assumptions. Timing attacks, such as side-channel attacks, do not break the underlying cryptographic primitives but instead target system-level behaviors or implementations. These attacks highlight the need to address practical security concerns alongside the theoretical strengths of HE.

Side-channel attacks exploit unintended information leakage during cryptographic operations, such as key generation or decryption. For instance, Aydin et al. demonstrated how the Number Theoretic Transform (NTT) in SEAL's key generation process could be vulnerable to such attacks, even when countermeasures like random delay insertion are applied [36]. However, our framework does not involve key generation, nor does it allow the server access to plaintexts. Similarly, the cache timing attack described in [37] leverages Barrett multiplication during decryption to infer secret information. Since our setup involves no decryption service on the server side, this class of attack is not applicable.

Timing attacks leverage variations in computation time to infer sensitive information. For example, the study of error leakage via timing channels in the TFHE library [38] shows how attackers can cluster timing data to reduce the range of error values, potentially enabling future recovery of secret keys. While this attack is relevant in HE schemes involving ciphertext decryption, it does not affect our system, as the server performs operations entirely on encrypted data. To further safeguard against timing attacks, we ensure constant-time responses by fixing the number and depth of trees in computations and adding dummy trees when necessary. These measures obfuscate any observable timing differences, rendering timing attacks infeasible in our framework.

B. LITERATURE OF SECURITY ATTACKS

ML systems, while powerful, are increasingly subject to various security threats that compromise their integrity, confidentiality, and availability. These attacks range from poisoning attacks that aim to corrupt the training process to inference attacks that exploit vulnerabilities in deployed models. Below, we explore key categories of ML security attacks and their relevance to PPML frameworks.

Poisoning attacks target the training phase of ML models, where adversaries inject malicious data into the training set to degrade the model's performance or manipulate its behavior. For instance, a subtle modification of the training data could bias a model toward incorrect outputs, enabling adversarial manipulation of predictions. While these attacks are significant, our work focuses on secure inference and does not involve training, which limits the applicability of poisoning attacks in our context. However, understanding these threats is critical for developing end-to-end PPML systems.

Membership inference attacks aim to determine whether a specific data point was part of a model's training dataset, posing significant privacy risks. These attacks typically analyze differences in the model's output probabilities or confidence scores, which may inadvertently reveal information about the data used during training. For instance, models trained on a specific dataset may exhibit overconfidence when predicting on data points they have encountered during training, providing adversaries with a statistical signal to infer membership.

In the context of PPML, mitigating membership inference attacks requires reducing overfitting and ensuring that the model's outputs reveal as little information as possible about the training data. Techniques such as differential privacy and regularization can help by minimizing overconfidence in predictions. While our framework does not specifically address membership inference attacks, the use of HE inherently adds a layer of protection by ensuring that all computations are performed on encrypted data. This prevents adversaries from directly accessing model outputs in plaintext, making it significantly harder to perform membership inference.

It is worth noting that while HE protects against direct exploitation of model outputs, a comprehensive defense against membership inference attacks in PPML frameworks would require incorporating additional techniques, such as

training with differential privacy, to reduce the risk of leakage from output distributions.

Other attack vectors in ML systems include evasion attacks, where adversaries craft adversarial examples to bypass the model, and model inversion attacks, which attempt to reconstruct sensitive information about the training data. While these attacks highlight the vulnerabilities of ML systems, our framework focuses on inference security under homomorphic encryption, where encrypted-only computations inherently mitigate many direct attacks on plaintext data. Nevertheless, addressing these threats in future research will be important for advancing the robustness of PPML systems.

C. THREAT MODEL

Our system operates under a client-server model, where the client encrypts its private data and sends it to the server for inference. The server hosts the model, which may either be encrypted or in plaintext. After performing inference on the client's encrypted data, the server returns the encrypted results to the client, who decrypts them locally. This setup ensures that sensitive data from both the client and the server remains secure during the process.

We assume a semi-honest adversarial setting, where the server and client are trusted to follow the protocol but may attempt to infer private information from the data they process. Specifically:

- Server's Perspective: The server may try to infer the client's private input from the encrypted data it receives or from intermediate computations during inference.
- Client's Perspective: The adversarial client may attempt to extract information about the server's model from the inference results or communication patterns.

We do not consider attacks where the server actively modifies the model or inference results, nor do we address tampering with the communication channel, as we assume it is secured by standard cryptographic means (e.g., TLS).

The primary assets we aim to secure include:

- Client's Private Data: The client's input data is encrypted before being sent to the server. Homomorphic encryption (HE) ensures that this data remains encrypted throughout the inference process, preventing the server from learning any plaintext information.
- Server's Model: The server's model parameters, whether encrypted or plaintext, are protected from the client. HE prevents the client from gaining access to intermediate computations or model details, safeguarding intellectual property and proprietary information.

Homomorphic encryption forms the cornerstone of our security model, providing a robust framework for privacy-preserving inference. Specifically, all client inputs remain encrypted during transmission and computation, preventing unauthorized access or leakage to the server. The server's model, if encrypted, is protected against any adversarial attempts by the client to extract its details. Since HE supports computations directly on encrypted data, no plaintext information is exposed at any point during inference. By combining these measures with a secure communication channel and assuming semi-honest participants, our system effectively mitigates key privacy and security risks, ensuring robust protection for both the client and the server

D. THE SECURITY ANALYSIS OF PRIVACY PRESERVING XGBoost INFERENCE

In our security analysis design, we build target models with the training data set, and the encoded model is stored in the server as suggested in [15]. The target model owner uses 16 split points for each variable in the feature set containing 196 features in total and uses random values to be used as dummy split points for the features with less than 16 split points. Note that all split points (real and dummy) are public to the model users. The attacker, which aims to build substitute models, encodes and then encrypts their training data as explained in Section IV and sends the resulting encrypted query to the model server. After receiving the encrypted model outputs, the attacker decrypts the results, which are then used as labels to build substitute models.

As explained in Section IV, the model server cannot disclose the encrypted user data, and the model users can only learn about the model by interpreting the outputs of the model. Therefore, we implement our security scenarios to build substitute models with a selected malicious user, the attacker. Furthermore, we assume that the malicious user (the attacker) does not have expert knowledge, therefore, uses all the features in the modeling phase as the particular subset of features used in the target model are not disclosed. The method used by the attacker for hyperparameter optimization is explained in Section V-D1. Also, the final substitute models and their performance metrics are given in Section V-D2. The attacker uses one (in scenario A) and two (in scenario B) of the six operation sites' data included in this study. Hence, we hypothesize that the attacker's training data does not have adequate variation to recreate the target model using only their own data set.

Even though the CKKS scheme is IND-CPA, the model leaks information with every output, which is revealed to the client, as mentioned previously. Hence, it is essential to analyze how much an adversary, acting as a legitimate client, might learn by querying the model. We conducted experiments to investigate the security of a particular model with respect to the number of queries granted to users. As explained in Section III-A, our data set consists of data points from six operation sites, each of which holds certain amount of genuine data points.

In scenario A, the data set of one (arbitrarily selected) site is kept for the use of the attacker, and the model (target) is built with the remaining data set. With scenario A, we simulate a real-world case whereby an attacker with a smaller data set queries the private model with its data, learns the results of the inference for all patients, and tries to reconstruct the target model or a model with comparable accuracy (*stealing the model*). In this scenario, 29% of the data set and 29%

of the mechanically complicated patients are reserved for the attacker.

In scenario B, data sets of two sites are kept for attacker use: 39% of the data set and 33% of the mechanically complicated patients. With scenario B, we simulate an attack when a coalition of users with a relatively larger and diverse set of data collude to steal the target model.

In the target model training phases, the data is split into training and test samples with a 75%-25% ratio for both scenarios. In the attack model training phase, training and test samples are selected to ensure that the test data has at least five mechanically complicated patients, that is, 75%-25% for scenario A and 68.75%-31.25% for scenario B. We use stratified sampling with respect to the presence of mechanical complications.

Finally, the attacker or the attacking coalition of users can produce synthetic data from their limited amount of genuine data to increase their chances of stealing the target model. Consequently, in each scenario, we have four attack options:

- Attack with the original attacking data set
- Attack with the original attacking data set + 50 synthetically generated data points
- Attack with the original attacking data set + 150 synthetically generated data points
- Attack with the original attacking data set + 600 synthetically generated data points

For generating the synthetic data set, we use Python's DataSynthesizer library to create synthetic data without the response variable [39]. Since we work on a correlated dataset, we initially want to use the correlated dataset mode, but the algorithm fails to build Bayesian networks. The authors in [39] suggest using the independent mode if the computation times are infeasibly high or the amount of data needs to be increased. Hence, we use the independent mode to create the synthetic data sets. The descriptive statistics of the substitute model training set of the important encoded features of the target model as well as the synthetic dataset's descriptive statistics are presented to assess the goodness of the synthetic datasets. Encoded values are presented instead of real values because the attacker queries the model with the encoded feature values. Furthermore, discrete two-sample Kolmogorov-Smirnov (KS) [40] test p-values are reported to assess if both sets are sampled from the same distribution.

The naïve method is chosen as the attack algorithm; we use the XGBoost algorithm to build attacker (substitute) models. In the rest of the paper, we refer to the models stolen by attackers as *substitute models*. As the clinical models must be sufficiently simple to consider in the surgery planning phases due to explainability, we elect to build markedly shallow tree-based models, which can make the target model especially vulnerable against attackers that can query the model with genuine and synthetic data points multiple times.

1) CROSS VALIDATION OF SUBSTITUTE MODELS

Two grid searches are conducted for each scenario using AUC as the performance metric since it is the suggested metric to use in the documentation of the XGBoost algorithm [6]. Again, nine times threefold cross-validation is used for hyperparameter optimization. When selecting the best-performing hyperparameter set, we pick the simplest ensemble with an AUC value, which can be at least the maximum AUC multiplied by validation size minus one divided by validation size. We deem an ensemble simpler than another if it has fewer trees, a higher gamma value, and greater minimum child weights.[2]

The means (and standard deviation) of the area under the receiving operating curve (AUROC), sensitivity, specificity, area under the precision-recall curve (AUPRC), precision, and F score are calculated for cross-validation analysis.

Note that we simulate the expert knowledge of the target model building phase by using only the seven features that are found significant in [20]. Since we assume that the attacker (the malicious user) does not have expert knowledge, the attacker uses all 196 features in the substitute model construction phase. To adjust the XGBoost algorithm to work with overly many features with a small data set, the attacker also tries to optimize the column (feature) subsample parameter.

2) MODEL OUTPUTS AND PERFORMANCE METRICS OF SUBSTITUTE MODELS

To better assess the substitute models' performance, the tuned hyperparameter set is used to construct 270 different models with ten seeds and 27 folds used in cross-validation. Models that perform best (with higher AUC) among those 270 models are used in the substitute model analysis. As we use the column subsample parameter in our hyperparameter tuning set, each node of the substitute models is chosen from a subset of all features. Hence we add randomness to the construction of the models and we use seeds to recreate the substitute models.

We summarize the models in three sections. In the first section, we create box plots of the accuracy, sensitivity, precision, and F-score metrics of the test set, where the accuracy figures of the validation sets are greater than 85%. In the second, we create box plots of the gain values of the features used by target models. Finally, in the third, we examine the split points of each feature of the substitute models and calculate their ratio to the split points of each feature used by the target models. This analysis is conducted using unique splits, meaning repetitions of the same split are counted as one.

Next, we report the differences in the prediction accuracy outcomes of the PPML and plaintext versions of the target models with the substitute models' training data as it is used by the attacker to build substitute models. Furthermore, the mean and standard deviations of the execution times of the preparation of a single query, PPML inference computation

[2]The gamma value is the minimum loss reduction allowed to create a new split, and the minimum child weight is the minimum sum of instance weight to create the child node.

for a single query, and decryption of the model output are reported, along with the file sizes of the encoded input and encrypted model output for a single query. Finally, we report the execution times of tree score calculations separately and ordering and node comparison operations together.

3) EXPERIMENTS WITH PUBLIC HEALTHCARE DATASETS

In [15] , Parkinson's [41] and Heart Disease [42] datasets are used to experiment with the proposed implementation in this study to assess the effects of the number of trees in an XGBoost model and the depth of the trees. Our proposed model suggests using dummy leaves and trees to match the lengths and depths of different models stored in the trusted server, ensuring that the timings are independent of such model features.

Execution times and accuracies of the PPML models are observed. The presented timings for these datasets are for the whole test sets to show that multiple queries at one time are possible. Twenty percent of the datasets are reserved as test sets.

VI. RESULTS

This section presents the security analysis scenarios' outputs and the execution times of privacy-preserving XGBoost inferences for a single query. The descriptive statistics of the features used by target models can be found in [20].

The classification and regression training (Caret) package of the RStudio is used for cross-validation and training of XGBoost models [43] alongside a data manipulation package, "dplyr" [44].

A. DESCRIPTIVE STATISTICS OF SUBSTITUTE MODEL TRAINING SETS

The means, standard deviations (SD), median values, and first and third-quartile values of the encoded substitute model training sets were presented in the first three columns of Table 2. The accompanying synthetics datasets' descriptive statistics were also given for scenario parts 2,3 and 4 in the remaining columns of Tables 2 and 3 in the same order, including p-values of the two-sample discrete Kolmogorov-Smirnov test results. Only the features in the target models were presented with the feature importance order of the target models. The original attacker training sets had 53 in scenario A and 66 patient records in scenario B. Since all the reported p-values are greater than or equal to 0.4, the null hypotheses that original and synthetic datasets were sampled from the same distribution were not rejected in each scenario part.

B. TARGET MODEL CROSS VALIDATION RESULTS

A single grid search was performed with 1500 parameter combinations for selecting the best parameter set with the lowest instrumented vertebra location (LIVLoc), estimated blood loss (EBL), pre-operative ODI walking score (ODI-ws), age, body mass index (BMI), the number of levels fused multiplied by construct location (FusebyConsLoc), and follow-up time after surgery (F-upTime) features. Five different numbers of trees ($3-7$), two different maximum depths ($2, 3$), five different learning rates ($0.1-0.5$), five different gamma values ($1-5$), and six different minimum child weights ($5-10$) parameters are tried. The positive scale ratio is set to 5, as used in [20]. Performance metrics of the cross-validation are presented in Table 4

The hyperparameters found for scenario A are 6 for the number of trees, 2 for maximum depth, 0.4 for learning rate, 4 for gamma, and 8 for minimum child weight parameters. For the scenario B, they are 5 for the number of trees, 2 for maximum depth, 0.4 for learning rate, 1 for gamma, and 6 for minimum child weight parameters.

C. TARGET MODEL PERFORMANCE METRICS

The final target models trained with found parameters and the whole training set produced six trees with 20 leaves in scenario A and five trees with 20 leaves in scenario B. The confusion matrices of the training and test data are presented in Table 5, and the accompanying metrics are in Table 6. Both target models had tree depths of two, producing four leaf nodes for each tree. Figure 3 represents all the trees of the target model in Scenario B.

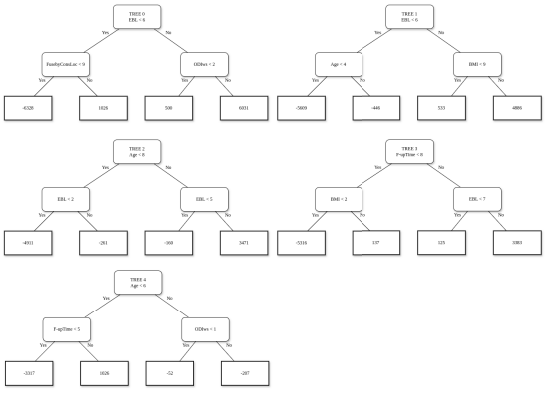


FIGURE 3. Encoded target model in Scenario B.

In Figure 4, the feature importance graphs for both scenarios are presented, where we can conclude that EBL is the most noteworthy feature in both scenarios.

The final tree ensemble built for scenario A has five (four unique) EBL split points, four (two unique) F-upTime split points, two (one unique) LIVLoc split points, two unique ODI-ws split points, and one BMI split point. In scenario B, the final ensemble has five (four unique) EBL split points, three (two unique) age split points, one FusebyConsLoc split point, two unique BMI split points, two unique ODI-ws split points, and two unique F-upTime split points.

D. ANALYSIS OF THE SUBSTITUTE MODELS

In this section, we investigate the substitute models constructed by attacker(s) using their real (and synthetic) data sets as queries to the target model in both scenarios.

TABLE 2. Descriptive statistics of substitute model encoded training sets (Part 1).

		Summary of original attacker training set		Summary of 50 synthetic data		
	Encoded Features (between 0-15)	Mean (SD)	First Quartile – Median – Third Quartile	Mean (SD)	First Quartile – Median – Third Quartile	KS test p-values
Scenario A	EBL	3.5 (4.0)	0 – 2 – 4	3.9 (4.0)	1 – 2 – 5.5	0.54
	F-upTime	5.5 (4.2)	2 – 4 -11	5.9 (3.9)	4 – 4 – 10.3	0.40
	LIVLoc	1.4 (2.0)	0 – 0 – 3	1.5 (2.0)	0 – 0 – 3	0.93
	ODI-ws	1.6 (1.5)	0 – 1 – 3	1.7 (1.4)	1 – 1 – 3	0.61
	BMI	5.2 (2.8)	3 – 5 – 6	5.5 (2.7)	4 – 5 – 6.75	0.71
Scenario B	EBL	5.5 (4.4)	2 – 4 – 9	5.9 (4.2)	2 – 4 – 9	0.57
	Age	5.5 (4.8)	2 – 4 – 10.8	6.0 (4.6)	2 – 4 – 10.5	0.74
	FusebyConsLoc	7.3 (3.3)	5 – 7 – 9	7.6 (3.2)	6.3 – 7 – 9.5	0.49
	BMI	4.6 (4.9)	1 – 3 – 8.8	4.7 (4.8)	1 – 3 – 8.8	0.75
	ODI-ws	1.6 (1.6)	0 – 1 – 3	1.9 (1.6)	1 – 2 – 3	0.43
	F-upTime	6.1 (4.8)	1 – 5 – 11	6.8 (4.6)	3 – 8 – 11	0.61

TABLE 3. Descriptive statistics of substitute model encoded training sets (Part 2).

		Summary of original attacker training set		Summary of 600 synthetic data		
	Encoded Features (between 0-15)	Mean (SD)	First Quartile – Median – Third Quartile	Mean (SD)	First Quartile – Median – Third Quartile	KS test p-values
Scenario A	EBL	3.6 (4.0)	0 – 2 – 4	3.6 (4.1)	0 – 2 – 4	1
	F-upTime	5.5 (3.9)	2 – 4 – 8	5.5 (4.2)	2 – 4 – 8	1
	LIVLoc	1.4 (1.9)	0 – 0 – 3	1.4 (2.0)	0 – 0 – 3	1
	ODI-ws	1.6 (1.5)	0 – 1 – 3	1.6 (1.5)	0 – 1 – 3	1
	BMI	5.2 (2.7)	3 – 5 – 6	5.2 (2.9)	3 – 5 – 6	0.99
Scenario B	EBL	5.7 (4.3)	2 – 6 – 8	5.5 (4.5)	2 – 4 – 9	1
	Age	5.7 (4.7)	2 – 4 – 9	5.6 (4.8)	2 – 4 – 10	1
	FusebyConsLoc	7.4 (3.3)	5 – 7 – 8	7.3 (3.5)	5 – 7 – 9	1
	BMI	4.5 (4.8)	1 – 3 – 8	4.6 (5.0)	1 – 3 – 8.25	1
	ODI-ws	1.7 (1.6)	0 – 2 – 3	1.7 (1.7)	0 – 1 – 3	1
	F-upTime	6.4 (4.7)	2 – 8 – 11	6.1 (4.8)	1 – 5 – 11	1

TABLE 4. Cross-validation results of target model building phases.

Scenario	AUC	TPR	TNR	AUPRC	Precision	F-score
A	0.72 (0.09)	0.53 (0.19)	0.79 (0.07)	0.21 (0.05)	0.28 (0.1)	0.37 (0.09)
B	0.73 (0.07)	0.58 (0.15)	0.79 (0.09)	0.32 (0.12)	0.39 (0.16)	0.45 (0.13)

Mean (standard deviations) are presented.
TPR and TNR represent true positive rate (sensitivity) and true negative rate (specificity), respectively.

TABLE 5. Confusion matrices of the training and test sets.

		Training Set		Test Set (Target)		Test Set (Attacker)	
Scenario	**Pred/Real**	**0**	**1**	**0**	**1**	**0**	**1**
A	**0**	102	2	33	1	9	1
	1	13	15	4	4	3	4
B	**0**	71	2	23	1	19	1
	1	21	17	7	6	6	4

0 represents healthy patients, and 1 represents patients with mechanical complications.

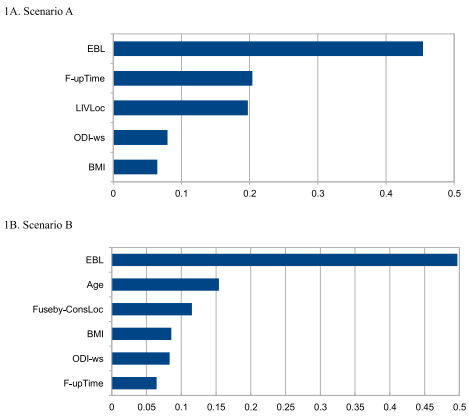


FIGURE 4. Feature importance graphs of target models for Scenario A and B.

1) CROSS-VALIDATION RESULTS

We performed two grid searches for each part of the security scenarios and presented the results in Table 7 and the accompanying hyperparameter set in Table 8. In Table 7, cross validation results are given as means and standard deviations (the latter in parenthesis). In both tables, each scenario is divided into four scenarios indicating whether synthetic data is also used in addition to real attack data. The number appended to the scenario letter (e.g., A1) indicates the number of synthetic data points used in the substitute model construction. The numbers 2, 3 and 4 stand for 50, 150 and 600 synthetic data points, respectively, while the number

TABLE 6. Performance metrics of the target model.

Scenario	Set	ACC	TPR	TNR	Precision	NPR	F-score
A	Training	0.89	0.88	0.89	0.54	0.98	0.67
	Test (Target)	0.88	0.80	0.89	0.50	0.97	0.62
	Test (Attacker)	0.76	0.80	0.75	0.57	0.90	0.67
B	Training	0.79	0.89	0.77	0.45	0.97	0.60
	Test (Target)	0.78	0.86	0.77	0.46	0.96	0.60
	Test (Attacker)	0.77	0.80	0.76	0.40	0.95	0.53

Mean (standard deviations) are presented.
ACC, TPR, and TNR represent accuracy, true positive rate (sensitivity) and true negative rate (specificity), respectively.

1 indicates only real data is used. Also, "**GS**" stands for grid search, where we apply two different sets of parameters. In the first one (i.e., GS1), we try 486 parameter combinations with 5, 10, and 15 trees, maximum depth of 2 and 3, learning rate of 0.1, 0.3 and 0.5, column subsample ratio of 0.6,0.8, and 1, and minimum child weight of 1,4, and 8. In the second parameter set used for the grid search (i.e., GS2), we operate in the proximity space of the tuned hyperparameter set used in GS1. On average, another 675 parameter combinations are explored with five different numbers of trees, three different learning rates, three different gamma values, three different column subsample ratios, and five different minimum child weights.

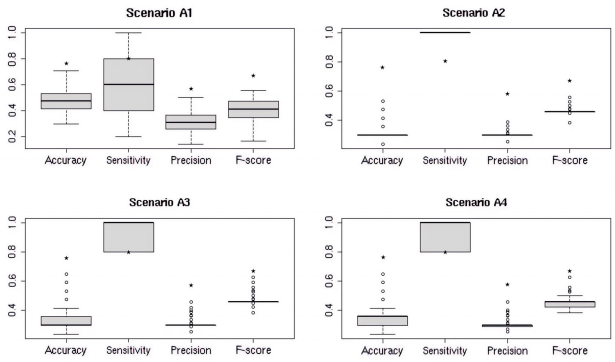
In Table 8, the scale positive weight is used to adjust the classification threshold. It is set to the ratio of the number of negative instances to positive instances.

As explained in Section III, we use a hyperparameter set with an AUC value, which is not always the best, to keep the model simple. We work with a hyperparameter set, whose accuracy can be as low as the one obtained when one patient is removed from the hyperparameter set, which gives the best accuracy. We also work with hyperparameter set resulting in fewer number of trees, a higher gamma value, and greater minimum child weights in the given order for selection criteria. For instance, in scenario B2, the hyperparameter set tuned in GS2 is simpler than GS1 as both have acceptably high AUC values, and there are fewer number of trees in the former.

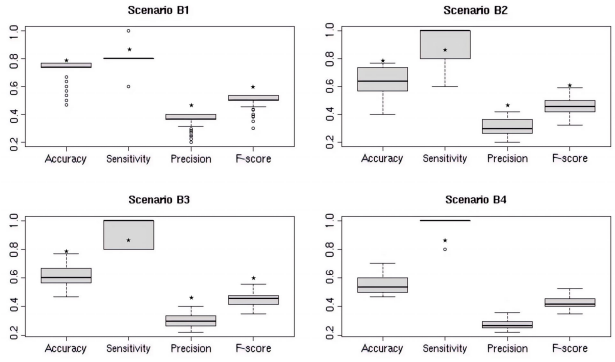
2) PERFORMANCE METRICS OF THE SUBSTITUTE MODELS

The performance metrics of the substitute models with validation accuracy greater than 85% are presented in Figures 5a and 5b, where 36, 125, 240, and 269 substitute models are plotted in Scenarios A1 to A4 respectively, while there are 265, 231, 198, and 27 substitute models in Scenarios B1 to B4, respectively. In the figures, the solid circles represent the performance metrics of the target models on the attacker test data. The figures illustrate all eight scenarios, which result in different number of substitute models. In scenario B4, the tuned column subsample ratio parameter is one according to cross-validation results. Therefore, for each seed, the same ensemble is built; in the end, we obtain 27 models instead of 270.

The best-performing attack of the scenario A queries the target model with only real attacking data. The means



(a) Scenarios A1 to A4



(b) Scenarios B1 to B4

FIGURE 5. Box plots of performance metrics of the substitute models on the attacker test set.

(standard deviations) of accuracy, sensitivity, precision, and F-score are 0.49 (0.12), 0.62 (0.25), 0.32 (0.09), and 0.41 (0.10), respectively. When synthetic data used in scenario A (scenarios A2-A4), the precision as well as accuracy figures are reduced. The mean (standard deviations) of accuracy are 0.31 (0.06), 0.36 (0.09), and 0.37 (0.11) for scenarios A2-A4, respectively. Similarly, the best-performing attack in the second scenario is B1, where the attacker queries the model with the real data points. The means (standard deviations) of accuracy, sensitivity, precision, and F-score are 0.73 (0.06), 0.80 (0.05), 0.36 (0.04), and 0.50 (0.05), respectively. The means (standard deviations) of accuracy values are 0.63 (0.09), 0.61 (0.07), and 0.0.55 (0.06) for scenarios B2-B4, respectively.

TABLE 7. Cross-validation results.

Scenario	AUC	Sensitivity	Specificity	AUPRC	Precision	F score
A1-GS1	0.87(0.09)	0.70(0.23)	0.86(0.08)	0.57(0.15)	0.70(0.11)	0.68(0.15)
A1-GS2	0.87(0.08)	0.74(0.21)	0.80(0.14)	0.52(0.13)	0.65(0.17)	0.68(0.16)
A2-GS1	0.90(0.06)	0.86(0.13)	0.83(0.08)	0.70(0.12)	0.78(0.08)	0.81(0.08)
A2-GS2	0.91(0.05)	0.90(0.07)	0.83(0.07)	0.63(0.11)	0.79(0.06)	0.84(0.05)
A3-GS1	0.97(0.02)	0.90(0.09)	0.92(0.05)	0.81(0.15)	0.89(0.06)	0.90(0.06)
A3-GS2	0.97(0.03)	0.90(0.09)	0.92(0.05)	0.77(0.21)	0.90(0.06)	0.90(0.06)
A4-GS1	0.99(0.00)	0.98(0.02)	0.97(0.04)	0.90(0.03)	0.96(0.04)	0.97(0.02)
A4-GS2	0.99(0.01)	0.95(0.06)	0.97(0.03)	0.76(0.26)	0.97(0.03)	0.96(0.03)
B1-GS1	0.99(0.01)	0.98(0.06)	0.92(0.09)	0.29(0.23)	0.95(0.06)	0.95(0.06)
B1-GS2	0.99(0.02)	0.99(0.05)	0.99(0.05)	0.21(0.20)	0.92(0.08)	0.95(0.05)
B2-GS1	0.94(0.04)	0.98(0.04)	0.75(0.15)	0.75(0.09)	0.90(0.06)	0.93(0.04)
B2-GS2	0.93(0.04)	0.98(0.04)	0.76(0.13)	0.45(0.19)	0.90(0.05)	0.94(0.03)
B3-GS1	0.96 (0.05)	0.95 (0.04)	0.77 (0.13)	0.81(0.09)	0.91(0.04)	0.93 (0.03)
B3-GS2	0.95(0.03)	0.95(0.04)	0.74(0.15)	0.70(0.14)	0.90(0.05)	0.92(0.03)
B4-GS1	0.99(0.01)	0.98(0.01)	0.90(0.07)	0.97(0.03)	0.97(0.02)	0.98 (0.01)
B4-GS2	0.99(0.01)	0.98(0.01)	0.90(0.07)	0.94(0.05)	0.97(0.02)	0.98(0.01)

Mean (standard deviations) are presented.
In scenario column, 1 represents the attack without synthetic data.
2,3 and 4 present attacks with 50, 150, and 600 synthetic data, respectively.

TABLE 8. Hyperparameter set tuned with cross validation.

Scenario	Scale positive weight	Num. of trees	Max. depth	Learning rate	Gamma	Column sub-sample	Min. child weight
A1-GS1	2.12	5	2	0.5	0	0.6	1
A1-GS2	2.12	5	2	0.5	0	0.5	3
A2-GS1	1.45	5	2	0.5	4	1	1
A2-GS2	1.45	5	2	0.6	3	0.8	2
A3-GS1	1.36	5	2	0.5	0	1	1
A3-GS2	1.36	4	2	0.6	2	0.9	1
A4-GS1	1.11	5	2	0.5	4	1	1
A4-GS2	1.11	4	2	0.5	2	0.8	4
B1-GS1	1	5	2	0.1	0	0.6	4
B1-GS2	1	4	2	0.1	2	0.7	4
B2-GS1	1	5	2	0.5	0	0.8	1
B2-GS2	1	3	2	0.5	3	0.7	1
B3-GS1	1	5	3	0.5	4	1	1
B3-GS2	1	4	3	0.5	5	0.9	1
B4-GS1	1	5	3	0.5	0	1	1
B4-GS2	1	4	3	0.5	0	1	1

The differences between performance metrics of best-substitute models and target models are 0.27, 0.18, 0.25, 0.26 for scenario A, and 0.04, 0, 0.04, and 0.03 for scenario B for accuracy, sensitivity, precision, and F-score, respectively (see Table 6). We discuss the reasons as to why scenario B performs well is discussed in Section VII. The differences between the accuracy value of the target model and the average accuracy values of all four scenarios on the substitute models' test data set are 0.38 for scenario A and 0.14 for scenario B, respectively. This shows that the substitute models cannot match the performance of the target model in both scenarios.

3) ANALYSIS OF FEATURE IMPORTANCE AND USED SPLIT POINTS

Even though the accuracy values obtained in Section VI-D2 imply that adding a synthetic data set to queries do not improve the substitute models' performance, the feature importance graphs can lead to different conclusion. As we work on shallow ensembles with a low number of trees, it can also be advantageous for an attacker to find out the features used by the target model. To assess the similarity of the feature importance of target and substitute models, we provide the box plots of the gain values of the features used by the target model in Figures 6 and 7. Again, substitute models with validation accuracy greater than 85% were used in this section and the solid circles represent the gain values of the target model.

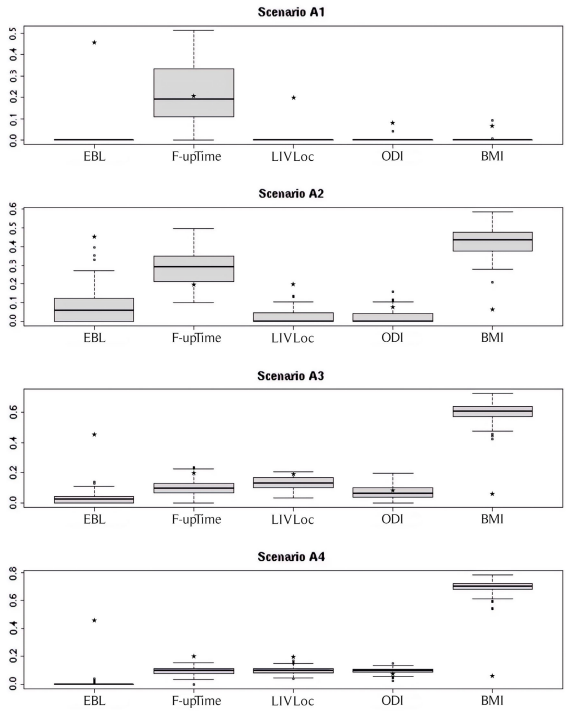


FIGURE 6. Box plots of gain values of the substitute models in attack Scenario A.

In scenario A1, out of 36 models, the F-upTime feature is used in 33 models, whereas LIVLoc and ODI-ws are used

in only two models. In scenario A2, 25% of the substitute models use the ODI-ws feature, 35% used the LIVLoc feature, and 50% used EBL. In all models, BMI and the F-upTime features are used. In scenario A3, EBL, ODI-ws, and F-upTime features are used in 53%, 94%, and 99% of the substitute models, respectively. BMI and LIVLoc features are used in all substitute models. In scenario A4, EBL and F-upTime features are used in 7% and 99% of the substitute models. LIVLoc, ODI-ws, and BMI features are used in all substitute models.

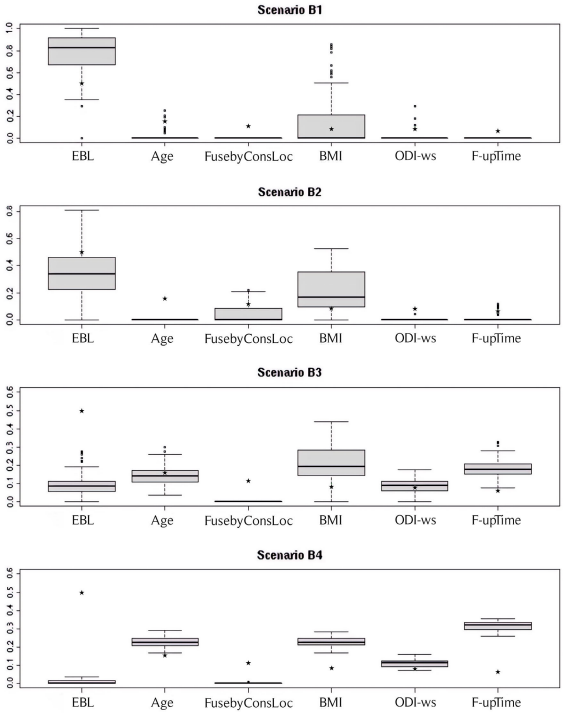


FIGURE 7. Box plots of gain values of the substitute models in attack Scenario B.

In scenario B1, out of 265 models, the EBL, BMI, age, and ODI-ws features are used in 98%, 48%, 4%, and 2% of the substitute models, respectively. In scenario B2, the signal from the age column is lost. EBL, BMI, FusebyConsLoc, and F-upTime features are used in 98%, 80%, 48%, and 19%, respectively. ODI-ws is used only once. In scenario B3, the signal from FusebyConsLoc is lost. On the other hand, the other five features employed by the target model are used in the majority of the substitute models. ODI-ws, BMI, and EBL features were in 84%, 94%, and 98% of the substitute models. Age and F-upTime features are used in all of the models. In scenario B4, out of 27 final substitute models, 2 used FusebyConsLoc feature, 13 of them used the EBL feature, and all 27 models used the other features used by the target model.

We compute two metrics for each scenario to assess the feature importance comparison results. The first metric is the average number of features common to both target and substitute models, and the second metric is the average of the sum of gain values of the substitute models of the features used both in the target and substitute models. In scenario A, the first metric measurements are 1.03 (21%), 3.10 (62%), 4.47 (89%), and 4.06 (81%) for scenario parts one to four, respectively. Percentages given in brackets are the ratio of this metric to the number of features used in the target model. The second metric measurements are 0.22, 0.83, 0.93, and 0.99 for scenario A, parts one to four, respectively. In scenario B, the first metric measurements are 1.51 (25%), 2.46 (41%), 4.76 (79%), and 4.56 (76%) for scenario parts one to four, respectively. The second metric measurements are 0.92, 0.61, 0.71, and 0.89 for scenario B, parts one to four. To assess the correct split point usages of the substitute models, we present the following heatmaps for each scenario in Table 9.

TABLE 9. Heatmaps of the percentages of target model split points used in substitute models.

Scenario	Validation Accuracy Threshold	Number of models	EBL	F-upTime	LIVLoc	ODI-ws	BMI
A1	0.85	36	0.00	0.01	0.00	0.00	0.00
A2	0.90	31	0.01	0.18	0.06	0.03	0.77
A3	0.95	45	0.04	0.06	1.00	0.50	0.78
A4	0.99	27	0.00	0.00	1.00	0.50	0.80

Scenario	Validation Accuracy Threshold	Number of models	EBL	Age	Fuseby ConsLoc	BMI	ODI-ws	F-up Time
B1	0.99	64	0.19	0.00	0.00	0.11	0.00	0.00
B2	0.95	43	0.26	0.00	0.09	0.22	0.00	0.01
B3	0.95	25	0.23	0.26	0.00	0.28	0.40	0.48
B4	0.98	3*	0.33	0.50	0.00	0.33	0.50	0.50

The validation accuracy thresholds are presented in the second column. The calculation is done using the unique usage of splits. In scenario B4, the tuned column subsample parameter in cross-validation was 1. Therefore, three models were selected with the threshold out of 27 final substitute models.

For each scenario, we summarize the correct usage of split points in the substitute models presented in Table 9.

In scenario A1, the F-upTime feature is used 33 times, and LIVLoc and ODI-ws twice. In four models, two target model features are used together. Only once the algorithm uses a target model split point of the F-up duration feature. In scenario A2, on average, three target model features are used. Substitute models use the target model's BMI split point correctly 24 times, one of the two F-upTime split points correctly 11 times, one of the two ODI-ws split points correctly twice, and one of the four EBL split points correctly once. In scenario A3, on average, 4.6 target model features are used. All substitute models use the target model's only LIVLoc split point correctly, and one of the two ODI-ws split points correctly. Substitute models use the target model's only BMI split point correctly thirty-five times, one of the four EBL split points correctly nine times, and one of the two F-upTime split points correctly five times. In scenario A4, on average, four target model features are used. All substitute models use the target model's only LIVLoc split point, and one of the two ODI-ws split points correctly. Twenty-one times, substitute models use the target model's only BMI split point correctly. On the other hand, none of the substitute

models use the target model's EBL and F-upTime split points correctly.

In scenario B1, on average, 1.6 target model features are used in substitute models. Forty-four times, substitute models use the target model's one of the four EBL split points correctly, and twice use two of the four split points correctly. Fourteen times substitute models use the target model's one of the two BMI split points correctly. In scenario B2, on average, 2.7 target model features are utilized in substitute models. Forty-two substitute models use the target model's one of the four EBL split points correctly, and once a substitute model uses two of the four split points correctly. Substitute models use the target model's one of the two BMI split points correctly nineteen times, FusebyConsLoc split point correctly four times, and one of the two F-upTime split points correctly only once. In scenario B3, on average, 4.9 target model features are utilized in substitute models. Substitute models use the target model's one of the two F-upTime split points correctly twenty-four times, one of the four EBL split points correctly twenty-three times, one of the two ODI-ws split points correctly twenty times, one of the two BMI split points correctly fourteen times, and one of the two age split points correctly thirteen times. In scenario B4, on average, five target model features are utilized in substitute models. In this scenario's final substitute model building phase, we use one as the column subsample parameter. Therefore, the different seeds produced the same models. The substitute models use the target model's one of the four EBL split points correctly twice, and the other used two. All substitute models use the target model's one of the two age, and one of the two ODI-ws split points correctly. The substitute models use the target model's one of the two BMI split points and one of the two F-upTime split points correctly, twice.

E. EXECUTION TIMES AND FILE SIZES

We test the PPML inferences of the target models by comparing the prediction results to the same public model. To compare the accuracies of the public and privacy-preserving XGBoost models, we use the training sets of the substitute models in each attack scenario with fifty-three and sixty-six patients in scenarios A, and B, respectively. In both cases, the privacy-preserving XGBoost model results are identical to the original model predictions.

In scenario A, the mean (standard deviation) duration to prepare a single encoded query is 2.68 (0.09) seconds. The mean (standard deviation) duration of encryption, including key generation, is 4.50 (0.05) seconds. There are three calculation steps before the final output, node comparison value calculation, tree score calculation, and final masking of the value. The mean (standard deviation) of the duration of each step are 0.17(0.01), 0.12(0.01), and 0.07 (0.01), respectively. The mean (standard deviation) duration of decryption in scenario A is 0.02 (0.001), and the mean (standard deviation) end-to-end duration of the whole process is 7.56 (0.11) seconds.

In scenario B, the mean (standard deviation) duration to prepare a single encoded query is 2.70 (0.06) seconds. The encryption's mean (standard deviation) duration, including key generation, is 4.48 (0.05) seconds. The mean (standard deviation) of the duration of node comparison calculation, tree score calculation, and final masking are 0.17 (0.01), 0.12 (0.02), and 0.07 (0.004), respectively. The mean (standard deviation) duration of decryption is 0.02 (0.001), and the mean (standard deviation) end-to-end duration of the whole process is 7.56 (0.08) seconds.

The encoded input and the encoded model outputs file sizes are 1.605 MB and 191 kB for both scenarios for each query.

To evaluate the practicality of our privacy-preserving XGBoost model, we compared its execution time to traditional, unencrypted inference methods. In our tests, the plain model (without encryption) generates predictions in less than 1 millisecond per query. In contrast, the privacy-preserving approach, which utilizes HE, experiences additional delays due to encryption, decryption, and operations on encrypted data.

In both scenarios, A and B, the average end-to-end duration for privacy-preserving inference is approximately 7.5 seconds per query, as measured on our current hardware. While these timings may improve with more powerful servers, they still lag behind those of the unencrypted model.

Despite being slower than the unencrypted version, the delay introduced by HE is reasonable for practical applications, such as healthcare, where privacy is of utmost importance. A response time of only a few seconds is still acceptable, especially given the strong privacy guarantees provided by HE. This comparison highlights that although HE adds computational complexity, it remains a viable option for use cases where privacy is a priority and the delay is manageable with adequate infrastructure.

F. EXPERIMENTS WITH PUBLIC HEALTHCARE DATASETS

The Heart Disease [42] dataset consists of 303 patients with 75 features and aims to predict the presence of heart diseases, whereas the Parkinson's [41] dataset consists of 197 patients with 23 features derived from biomedical voice measurements. In both cases, it is a binary classification problem, similar to predicting mechanical complications.

Reference [15] shows the difference in timing when the depth of the ensemble is changed from 3 to 5 using the Parkinson's dataset. Seventeen features are used in both models with 48 trees. The minimum encoding length for the queries is 8, whereas the minimum batch size and the number of rotations needed are 384 and 12, respectively.

The accuracies of the privacy-preserved Parkinson's models, P1 and P2, are 90% and 93%. On the contrary, the plain versions of the model achieve an accuracy of 93%. Model P1 has a relative error of 3%.

Reference [15] illustrates the difference in timing when the number of trees in the ensemble is varied, using 24, 48, and 128 trees in the ensemble with the Heart Disease dataset. Thirteen features are used in the final models, H1

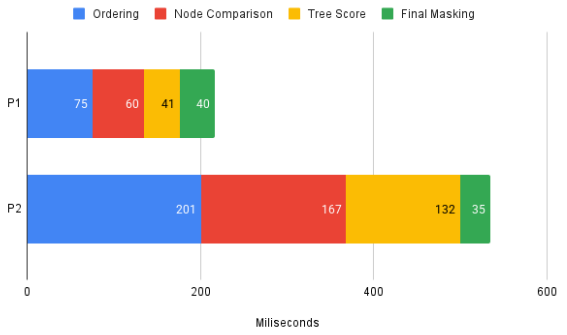


FIGURE 8. Execution times of privacy preserving Parkinson's models.

to H3, where the depth of the ensemble is 2. The minimum length of the encoding increases with the number of trees, starting at 4 and increasing to 8 and 16 for ensembles with 48 and 128 trees, respectively. Furthermore, the minimum batch sizes and rotations needed are 96, 384, and 2048, and 9, 12, and 15 for models H1 to H3, respectively.

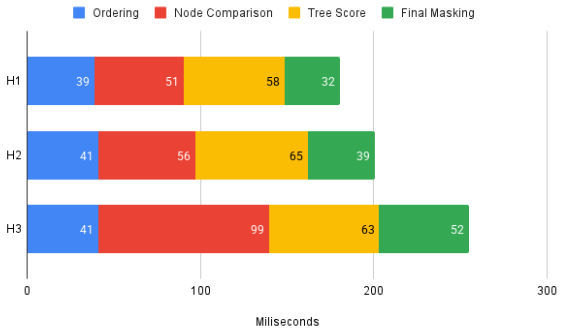


FIGURE 9. Execution times of privacy preserving heart disease models.

The accuracies of the privacy-preserved Heart Disease models, H1 to H3, are 89%, 90%, and 89%. In contrast, the plain versions of the models achieve accuracies of 90%, 92%, and 89% for models H1 to H3, respectively. The relative error is 1% for the H1 model and 2% for the H2 Heart Disease model.

VII. DISCUSSION

Our work is designed to investigate the possibility of using the ML models of the European Spine Study Group (ESSG) while protecting the intellectual property of the models and preventing the leakage of sensitive information from user data and the plaintext models, which can potentially expose sensitive training data. We aim to find an acceptable query limit for the other ASD surgery sites willing to benefit from the model while the security of both the model and the query data is preserved. The security of the CKKS scheme we used in the implementation is IND-CPA; hence the safety of the input data is held. Yet, with each query and the encoding method itself, we are giving away some information about the model with the query results.

Various algorithms have been developed in recent years to 'steal' a model with the query data of a model. However, these works mainly focused on complex models such as neural networks. The two examples of such models are Copycat CNN [45] and Knockoff Nets [46], where both methods construct neural networks. In our study, we worked only with the naïve method (the XGBoost algorithm) to attack as we attempted to mimic the less complex target ensembles in the attack section.

We have two issues regarding the weaknesses of our study that we want to discuss. First, even though the attacker data is split prior to building a model to be attacked (target), the seven features used in the modeling phase are derived from the whole dataset following the work [20]. This would increase the chance of a successful attack as the attacker data contributes to the feature selection method of the target model-building phase. Second, we had to use the synthetic data in the validation sets for scenario parts two to four alongside the original dataset. This might contribute to the loss of signals comparing parts 3 and 4 in both scenarios. The average number of features used by target and attacker models reduces from 4.47 to 4.06 and 4.76 to 4.56 in scenarios A and B, parts 3 and 4, respectively. In Figure 4, we observe some lost signals, even though the trend to use the target model split points as we increase the query size with synthetic data is visible.

One of the main technical challenges is the significant computational overhead introduced by homomorphic encryption, particularly when performing operations necessary for tree-based ensemble inferences, such as comparisons at each node of the model trees [28], [29], [47]. The HE scheme we used also needed to support summation as we needed to sum all the tree scores, yet the bottleneck of the HE schemes is comparison functions. Next, since the HEAAN library uses approximation, we expected possible differences between the accuracies of the PPML and public models. Yet, because our ensembles are relatively shallow, no differences were observed. Hence, there was no information loss due to the usage of approximation of the HEAAN library. This achievement indicates that our PPML strategies and implementations were effectively designed to handle the complexities associated with operating on encrypted data. By ensuring that our encryption and privacy-preserving techniques did not interfere with the core functionalities and algorithms of the XGBoost models, we preserved the integrity and accuracy of predictions. This success suggests that it is possible to achieve high levels of data security without sacrificing the performance of the model, challenging the common trade-off scenario often encountered in PPML applications.

There was not enough evidence to conclude that means of total timings differ by comparing the timing results of both scenarios with the homoscedastic two-sample t-test. (p-value 0.9) Yet the mean duration of encryption computations varies (p-value 0.004), where the encryption process took 0.2 seconds longer in scenario A on average.

This might be the effect of one more tree in the ensemble. As [25] suggested, further investigation is needed to pinpoint the reason for the difference. We may overcome this problem by fixing the tree size by adding dummy trees until the number of trees reaches a predefined maximum number.

Advantages of using our proposed privacy-preserving XGBoost, or any tree-based ensemble, inference algorithm are no loss of accuracy with the ESSG dataset and feasible timing results, as explained in the previous two paragraphs. On the other hand, as we showed in the results section, if there is no query limit the target models are susceptible to leak sensitive information. Therefore, for each model, an extensive security analysis is needed after the model development process. This is the main disadvantage of our proposed algorithm.

Even though there is no accuracy loss when using the ESSG dataset, our implementation produced acceptable relative errors (0–3%) in experiments with the Parkinson's and Heart Disease datasets. This indicates that accuracy loss is dependent on the dataset and the model itself. The timing results of these models show that the depth of the trees is a crucial factor in execution times. The execution time increases by a factor of 2.5 when the depth of the trees is increased by two levels. Additionally, an increase in the number of trees in a model results in a slight increase in execution time. This is due to the increase in the number of split values for a feature, which leads to an increase in both the encoding length and the number of rotations. These experiments also demonstrate that our proposed method can be applied to other real-world problems, taking into account the relative error for each specific scenario. The proposed method is scalable, even when an abundant number of dummy trees and nodes are added, as evidenced by the timing results of the ESSG dataset with plaintext models.

Since in proportionally aligned ASD patients according to the GAP scoring methodology, it is more critical to detect mechanical complications to treat them proactively. We build highly sensitive models, resulting in low-precision target models. (See Table 3) This leads to the decrease in the ratio of the number of healthy patients over the number of patients with mechanical complications diminishes from 4.9 (in the whole dataset) to 2.1 and 1 in the original attacker training set of scenarios A and B, respectively. Adding the synthetic dataset to the query shifts this balance in favor of increasing the number of mechanical complication predictions, as seen in the decrease of scale positive weight parameter in Table 5, which is the abovementioned ratio. Even though the attacker cross-validation mean precisions are between 65% and 97%, attacker models fail to achieve such precisions in the test set because of this shift in both scenarios, parts 2-4.

Even though using more synthetic data does not increase the performance metrics of the attacks, by logic, more queries result in more information leakage, hence the risk of disclosing the models. We see this when we analyze the gain values of the attacker models. We observe a remarkable difference between parts 2 and 3 in both scenarios' average target model features used in attacker models, increasing from 62% to 89% and 41% to 79% in scenarios A and B, respectively. Using part 3 results, attackers may adapt new data generation methods by altering those features. This would increase the information leakage from the queried models. The heatmaps in Figure 4 validate our decision because we see usages of correct split points using 150 synthetic data to the original dataset.

While our homomorphic encryption–based approach robustly secures both patient data and model intellectual property, it still faces several nuanced limitations. First, although tree-based models are computationally friendlier than deep neural networks, scaling to significantly deeper or extremely large ensembles may increase ciphertext size and polynomial-level operations, raising challenges in real-world clinical environments where response times matter.Another concern relates to clinical interpretability: despite using a family of models (decision trees) that are typically transparent, the intermediate logic remains sealed under encryption, which may complicate compliance requirements or second-opinion audits. Finally, we rely on external cryptographic libraries that evolve rapidly, posing extra hurdles for stable, long-term healthcare deployments. Addressing these limitations, especially in high-stakes medical contexts, is crucial for the widespread adoption of privacy-preserving ML systems.

VIII. FUTURE WORK

Future research may address to enhance the scope and applicability of this study. Expanding the number of participant sites and increasing dataset sizes could improve the generalizability of the findings and allow for more robust statistical analyses. Larger and more diverse datasets would better capture variability in real-world applications, leading to a more comprehensive evaluation of the proposed methodologies. Poisoning attacks can be addressed in a federated learning study as a future work, since this research focused on the inference phase of the modeling.

Our framework also shares its security assumptions with lattice-based cryptographic constructions, which are the foundation of new post-quantum signature schemes. For example, Raccoon [48], a module-based signature scheme built on the Module Learning With Errors (MLWE) problem, has shown promise in achieving post-quantum security while being resilient to side-channel attacks. Exploring how advancements in PQC signature schemes can complement or enhance the security of HE-based frameworks could further contribute to the development of secure and efficient PPML systems, particularly as these technologies continue to advance in sensitive applications such as healthcare.

IX. CONCLUSION

To conclude, it would be safe to allow 100 queries following the assessment in the discussion. In part 2 of

both scenarios, more than 100 queries are used. In both scenarios, 50 synthetic data is queried along with the original attacker dataset, excluding the test data. 103 and 116 times, the ensemble is queried in parts two of scenarios A and B, respectively. The maximum patient number of the sites that participated and the timespan of the surgeries in this work are 70 patients and ten years, respectively. Hence the query limit would be sufficient for 14 years, referencing the mentioned site. Last, a single query's timings and file sizes to the privacy-preserving model inference are adequate for practical usage.

First, while Parra-Ullauri et al. [7] successfully encrypt data during training, they do not thoroughly address inference-specific threats such as model-stealing. By design, our protocol includes query-based protections that limit adversaries' ability to reverse-engineer the model, thereby safeguarding the model's IP. Second, although Lee et al. [8] and Chen et al. [9] employ homomorphic encryption for healthcare applications, they place heavier emphasis on protecting data holders and often report more significant slowdowns. In contrast, we retain near-identical model metrics (identical in the ESSG dataset) and manageably short inference times, thus achieving a more practical runtime profile for real clinical environments. Third, approaches using differential privacy—for example, Wei et al. [10] — can provide robust formal guarantees across distributed or federated systems but may degrade final model performance due to injected noise. Finally, while Wu et al. [11] and Hassan et al. [12] address tree-based ensembles under encryption, their work primarily secures patient data alone. Our method provides additional query-limiting mechanisms, preventing adversaries from gleaning critical threshold splits or leaf values that would expose the model's design. This more holistic approach strengthens end-to-end security—covering not only data privacy but also the intellectual property of the model in high-stakes clinical settings. By focusing on inference protection, we pave the way for broader deployment of third-party analytics in healthcare, ensuring that both patients and model owners remain safe from advanced threats.

The major limitation of this study is the need for well-organized adult spinal deformity datasets. We overcome this by using synthetic datasets to increase the possible substitute model queries. Both weaknesses discussed in the discussion section are the consequence of using a relatively small dataset compared to other (privacy-preserving) machine learning studies.

We, as the authors, believe that such machine learning models (i.e., our target models) may contribute to the success of the related industry. When the security of both the model and the user data is secured, their usage will give an incentive to share and query such ML models.

X. DECLARATIONS

CONSENT FOR PUBLICATION

Our manuscript does not contain any individual person's data in any form.

AVAILABILITY OF DATA AND MATERIALS

The datasets generated and analyzed during the current study are not publicly available due to the clauses of the ESSG Association and ESSG Database Policy. Still, they are available from the corresponding author on reasonable request.

COMPETING INTERESTS

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Ahmet Alanay reports a relationship with Medtronic that includes: funding grants. Ahmet Alanay reports a relationship with DePuy Synthes that includes: funding grants. Ahmet Alanay reports a relationship with Globus Medical Inc. that includes: consulting or advisory. Ahmet Alanay reports a relationship with ZimVie that includes: royalties and consulting. Caglar Yilgor reports a relationship with Medtronic that includes: consulting. Javier Pizones reports a relationship with Stryker that includes: consulting. Javier Pizones reports a relationship with Medtronic that includes: consulting and funding grants. Javier Pizones reports a relationship with DePuy Synthes that includes: funding grants. Frank Kleinstueck reports a relationship with DePuy Synthes that includes: speaking and lecture fees. Ibrahim Obeid reports a relationship with Alpatec Spine that includes: royalties. Ibrahim Obeid reports a relationship with Medicrea that includes: royalties. Ibrahim Obeid reports a relationship with Spineart that includes: royalties. Ibrahim Obeid reports a relationship with DePuy Synthes that includes: consulting and funding grants. Ibrahim Obeid reports a relationship with Medtronic that includes: consulting. Ferran Pellisé reports a relationship with Medtronic that includes: consulting and funding grants. Ferran Pellisé reports a relationship with Stryker Spine that includes: consulting. Ferran Pellisé reports a relationship with DePuy Synthes that includes: funding grants. European Spine Study Group reports a relationship with Medtronic that includes: funding grants. European Spine Study Group reports a relationship with DePuy Synthes that includes: funding grants. Baris Balaban declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article. Seyma Selcan Magara declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article. Altug Yucekul declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article. Francisco Javier Sanchez Perez-Grueso declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article. Erkay Savas declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article. Çetin Bağcı declares that he has no known competing

financial interests or personal relationships that could have appeared to influence the work reported in this article. Osman Ugur Sezerman declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

AUTHORS' CONTRIBUTIONS

Baris Balaban contributed to the conception and design, analysis and interpretation of data, drafting of the manuscript, implementation, and statistical analysis. Seyma Selcan Magara contributed to the conception and design, drafting of the manuscript, and implementation. Caglar Yilgor contributed to the conception and design, data acquisition, interpretation of data, and administrative support. Altug Yucekul contributed to conception and design, and data acquisition. Ibrahim Obeid contributed to conception and design, and data acquisition. Javier Pizones contributed to conception and design, and data acquisition. Frank Kleinstueck contributed to conception and design, and data acquisition. Francisco Javier Sanchez Perez-Grueso contributed to conception and design, and data acquisition. Ferran Pellisé contributed to conception and design, and data acquisition. Ahmet Alanay contributed to conception and design, and data acquisition. Erkay Savas contributed to conception and design, critical revision of the manuscript for important intellectual content, and technical support. Çetin Bağcı contributed to administrative support. Osman Ugur Sezerman contributed to administrative support and critical revision of the manuscript for important intellectual content. All authors read and approved the final manuscript.

EUROPEAN SPINE STUDY GROUP

Caglar Yilgor, Altug Yucekul, and Ahmet Alanay are with the Department of Orthopedics and Traumatology, Acibadem University School of Medicine, Istanbul, Türkiye. Ibrahim Obeid is with Clinique du Dos, Elsan Jean Villar Private Hospital, Bordeaux, France. Javier Pizones and Francisco Javier Sanchez Perez are with the Spine Surgery Unit, Hospital Universitario La Paz, Madrid, Spain. Frank Kleinstueck is with the Spine Center Division, Department of Orthopedics and Neurosurgery, Schulthess Klinik, Switzerland. Ferran Pellisé is with the Spine Surgery Unit, Hospital Universitari Vall d'Hebron, Barcelona, Spain.

ACKNOWLEDGMENT

Baris Balaban is a bursar of the Scientific and Technological Research Council of Turkey (TUBİTAK) 2244–Industry Doctorate Program in conjunction with Bilmed Computer and Software Company. Prof. Osman Ugur Sezerman supervised this work as the Department Head of the Bioinformatics and Biostatistics Department, Acibadem University, Türkiye, in conjunction with Prof. Erkay Savas from the Computer Engineering Department, Sabancı University. Çetin Bağcı was the industry supervisor of this work.

The European Spine Study Group (ESSG) was founded in July 2010 by a group of European spinal deformity surgeons who decided to develop a comprehensive, prospective, multicentre, European and international adult spinal deformity (ASD) database to evaluate the clinical outcomes of patients with ASD undergoing conservative or surgical treatment. Funds and group administration were centralized in the Vall Hebron Institut de Recerca (VHIR) Research Foundation, Barcelona.

By January 2012, the first administrative and legal group structure with four centers (Ankara Spine Center in Ankara, Vall d'Hebron Hospital in Barcelona, Acıbadem University School of Medicine in Istanbul, Türkiye, and La Paz University Hospital in Madrid) was established, IRB approval was obtained for all centers, and a central study coordinator plus local research coordinators for each center were selected. In September 2013, the group attained its present structure, which includes two additional sites (Bordeaux University Hospital and the Schulthess Klinik in Zürich).

The final goal of ESSG is to improve the medical care and health related quality of life of patients with ASD by performing high quality research and delivering valuable knowledge through the analysis and mining of the ESSG database. A large part of our research funds are used to ensure the acquisition of high quality data, with the data collection and entry being managed by our well-trained, local research coordinators. A central coordinator supervises and coordinates the activities of all the sites involved in the study group and ensures that professional research standards are maintained throughout.

REFERENCES

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aaa8415

[2] K. E. Lauter, "Private AI: Machine learning on encrypted data," Cryptol. ePrint Archive, Paper 2021/324, 2021. [Online]. Available: https://eprint.iacr.org/2021/324

[3] K. Tiwari, S. Shukla, and J. P. George, "A systematic review of challenges and techniques of privacy-preserving machine learning," in *Data Science and Security*, S. Shukla, A. Unal, J. V. Kureethara, D. K. Mishra, and D. S. Han, Eds., Singapore: Springer, 2021, pp. 19–41.

[4] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, New York, NY, USA, May 2009, pp. 169–178, doi: 10.1145/1536414.1536440.

[5] Y. Ameur, S. Bouzefrane, and V. Audigier, "Application of homomorphic encryption in machine learning," in *Emerging Trends in Cybersecurity Applications*, K. Daimi, A. Alsadoon, C. Peoples, and N. E. Madhoun, Eds., Cham, Switzerland: Springer, Jan. 2023, pp. 391–410. [Online]. Available: https://hal.science/hal-0393330

[6] X. Meng and J. Feigenbaum, "Privacy-preserving XGBoost inference," 2020, *arXiv:2011.04789*.

[7] J. M. Parra-Ullauri, L. F. Gonzalez, A. Bravalheri, R. Hussain, X. Vasilakos, I. Vidal, F. Valera, R. Nejabati, and D. Simeonidou, "Privacy preservation in kubernetes-based federated learning: A networking approach," in *Proc. IEEE IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2023, pp. 1–7.

[8] J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim, and J.-S. No, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30039–30054, 2022.

[9] X. Chen, R. Li, and Y. Zhou, "Secure gradient boosting for medical diagnosis with partially homomorphic encryption," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2023, pp. 1–12.

[10] Y. Wei, A. T. Tran, and H. Wang, "Federated XGBoost for privacy-preserving telemedicine: A differential privacy approach," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1234–1245, 2023.

[11] L. Wu, K. Zhang, and M. Xu, "Encrypted decision tree ensembles for sensitive healthcare data," *ACM Comput. Surveys*, vol. 55, no. 10, pp. 1–30, 2022.

[12] M. Hassan, M. A. Butt, and M. Zaman, "An ensemble random forest algorithm for privacy preserving distributed medical data mining," *Int. J. E-Health Med. Commun. (IJEHMC)*, vol. 12, no. 6, pp. 1–23, Nov. 2021.

[13] F. J. Jaime, A. Muñoz, F. Rodríguez-Gómez, and A. Jerez-Calero, "Strengthening privacy and data security in biomedical microelectromechanical systems by IoT communication security and protection in smart healthcare," *Sensors*, vol. 23, no. 21, p. 8944, Nov. 2023.

[14] A. Muñoz, R. Ríos, R. Román, and J. López, "A survey on the (in)security of trusted execution environments," *Comput. Secur.*, vol. 129, Jun. 2023, Art. no. 103180.

[15] S. S. Magara, "Privacy-preserving XGBoost inference with homomorphic encryption," M.S. thesis, Dept. Comput. Sci. Eng., Sabanci Univ., İstanbul, Turkey, 2022.

[16] Q. Santos, "Cryptography for pragmatic distributed trust and the role of blockchain," École Normale Supérieure, PSL Res. Univ., Paris, France, Tech. Rep. tel-01966109, 2018.

[17] K. Lauter, "Private AI: Machine learning on encrypted data," in *Recent Advances in Industrial and Applied Mathematics*, T. Chacón Rebollo, R. Donat, and I. Higueras, Eds., Cham, Switzerland: Springer, 2022, pp. 97–113.

[18] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001, doi: 10.1214/aos/1013203451.

[19] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," 2016, *arXiv:1603.02754*.

[20] B. Balaban, C. Yilgor, A. Yucekul, T. Zulemyan, I. Obeid, J. Pizones, F. Kleinstueck, F. J. S. Perez-Grueso, F. Pellise, A. Alanay, and O. U. Sezerman, "Building clinically actionable models for predicting mechanical complications in postoperatively well-aligned adult spinal deformity patients using XGBoost algorithm," *Informat. Med. Unlocked*, vol. 37, Jan. 2023, Art. no. 101191. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914823000333

[21] A. Grivet Sébert, "Combining differential privacy and homomorphic encryption for privacy-preserving collaborative machine learning," Doctoral thesis, Dept. Comput. Sci. Digit. Sci., Université Paris-Saclay, Orsay, France, Jun. 2023. [Online]. Available: https://theses.hal.science/tel-04223076

[22] R. Gupta, D. Saxena, I. Gupta, and A. K. Singh, "Differential and TriPhase adaptive learning-based privacy-preserving model for medical data in cloud environment," *IEEE Netw. Lett.*, vol. 4, no. 4, pp. 217–221, Dec. 2022.

[23] R. Gupta and A. K. Singh, "A differential approach for data and classification service-based privacy-preserving machine learning model in cloud environment," *New Gener. Comput.*, vol. 40, no. 3, pp. 737–764, Jul. 2022, doi: 10.1007/s00354-022-00185-z.

[24] A. K. Singh and R. Gupta, "A privacy-preserving model based on differential approach for sensitive data in cloud environment," *Multimedia Tools Appl.*, vol. 81, no. 23, pp. 33127–33150, Apr. 2022, doi: 10.1007/s11042-021-11751-w.

[25] Ş. S. Mağara, C. Yıldırım, F. Yaman, B. Dilekoğlu, F. R. Tutaş, E. Öztürk, K. Kaya, Ö. Taştan, and E. Savaş, "ML with HE: Privacy preserving machine learning inferences for genome studies," 2021, *arXiv:2110.11446*.

[26] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, Oct. 2006, pp. 79–88, doi: 10.1145/1180405.1180417.

[27] M. Chase and S. Kamara, "Structured encryption and controlled disclosure," in *Advances in Cryptology—ASIACRYPT 2010*. Berlin, Germany: Springer, 2010, pp. 577–594.

[28] E. Lee, J.-W. Lee, J.-S. No, and Y.-S. Kim, "Minimax approximation of sign function by composite polynomial for homomorphic comparison," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 6, pp. 3711–3727, Nov. 2022. [Online]. Available: https://eprint.iacr.org/2020/834

[29] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology—ASIACRYPT 2017*, T. Takagi and T. Peyrin, Eds., Cham, Switzerland: Springer, 2017, pp. 409–437.

[30] M. Babenko, A. Tchernykh, E. Golimblevskaia, L. B. Pulido-Gaytan, and A. Avetisyan, "Homomorphic comparison methods: Technologies, challenges, and opportunities," in *Proc. Int. Conf. Eng. Telecommun.*, Nov. 2020, pp. 1–5.

[31] C. Yilgor, N. Sogunmez, Y. Yavuz, L. Boissiere, I. Obeid, E. Acaroglu, A. F. Mannion, F. Pellise, and A. Alanay, "Global alignment and proportion (GAP) score: Development and validation of a new method of analyzing spinopelvic alignment to predict mechanical complications after adult spinal deformity surgery," *Spine J.*, vol. 17, no. 10, pp. S155–S156, Oct. 2017.

[32] S. H. Noh, Y. Ha, I. Obeid, J. Y. Park, S. U. Kuh, D. K. Chin, K. S. Kim, Y. E. Cho, H. S. Lee, and K. H. Kim, "Modified global alignment and proportion scoring with body mass index and bone mineral density (GAPB) for improving predictions of mechanical complications after adult spinal deformity surgery," *Spine J.*, vol. 20, no. 5, pp. 776–784, May 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1529943019310927

[33] J. S. Smith, C. I. Shaffrey, V. Lafage, F. Schwab, J. K. Scheer, T. Protopsaltis, E. Klineberg, M. Gupta, R. Hostin, K.-M. G. Fu, G. M. Mundis, H. J. Kim, V. Deviren, A. Soroceanu, R. A. Hart, D. C. Burton, S. Bess, C. P. Ames, and I. S. S. Group, "Comparison of best versus worst clinical outcomes for adult spinal deformity surgery: A retrospective review of a prospectively collected, multicenter database with 2-year follow-up: Presented at the 2015 AANS/CNS joint section on disorders of the spine and peripheral nerves," *J. Neurosur., Spine*, vol. 23, no. 3, pp. 349–359, Sep. 2015. [Online]. Available: https://thejns.org/spine/view/journals/j-neurosurg-spine/23/3/article-p349.xml

[34] F. Pellisé, M. Serra-Burriel, A. Vila-Casademunt, J. L. Gum, I. Obeid, J. S. Smith, F. S. Kleinstück, S. Bess, J. Pizones, V. Lafage, F. J. S. Pérez-Grueso, F. J. Schwab, D. C. Burton, E. O. Klineberg, C. I. Shaffrey, A. Alanay, and C. P. Ames, "Quality metrics in adult spinal deformity surgery over the last decade: A combined analysis of the largest prospective multicenter data sets," *J. Neurosurg., Spine*, vol. 36, no. 2, pp. 226–234, Feb. 2022. [Online]. Available: https://thejns.org/spine/view/journals/j-neurosurg-spine/aop/article-10.3171-2021.3.SPINE202140/article-10.3171-2021.3.SPINE202140.xml

[35] E. Quarto, A. Zanirato, M. Pellegrini, S. Vaggi, F. Vitali, S. Bourret, J. C. Le Huec, and M. Formica, "GAP score potential in predicting post-operative spinal mechanical complications: A systematic review of the literature," *Eur. Spine J.*, vol. 31, no. 12, pp. 3286–3295, Dec. 2022, doi: 10.1007/s00586-022-07386-6.

[36] F. Aydin and A. Aysu, "Leaking secrets in homomorphic encryption with side-channel attacks," *J. Cryptograph. Eng.*, vol. 14, no. 2, pp. 241–251, Jun. 2024.

[37] W. Cheng, J.-L. Danger, S. Guilley, F. Huang, A. B. Korchi, and O. Rioul, "Cache-timing attack on the seal homomorphic encryption library," in *Proc. 11th Int. Workshop Secur. Proofs Embedded Syst.*, Sep. 2022, pp. 1–16. [Online]. Available: https://api.semanticscholar.org/CorpusID:269433046

[38] B. Chaturvedi, A. Chakraborty, A. Chatterjee, and D. Mukhopadhyay, "Error leakage using timing channel in FHE ciphertexts from TFHE library," Cryptol. ePrint Archive, Paper 2022/685, 2022. [Online]. Available: https://eprint.iacr.org/2022/685

[39] H. Ping, J. Stoyanovich, and B. Howe, "DataSynthesizer: Privacy-preserving synthetic datasets," in *Proc. 29th Int. Conf. Sci. Stat. Database Manage.*, New York, NY, USA, Jun. 2017, pp. 1–5, doi: 10.1145/3085504.3091117.

[40] D. S. Dimitrova, V. K. Kaishev, and S. Tan, "Computing the Kolmogorov–Smirnov distribution when the underlying CDF is purely discrete, mixed, or continuous," *J. Stat. Softw.*, vol. 95, no. 10, p. 1, 2020. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v095i10

[41] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. Costello, and I. M. Moroz, "Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection," *Biomed. Eng. OnLine*, vol. 6, no. 1, p. 23, 2007, doi: 10.1186/1475-925x-6-23.

[42] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[43] M. Kuhn, "Building predictive models in R using the caret package," *J. Stat. Softw.*, vol. 28, no. 5, p. 1, 2008. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v028i05

[44] H. Wickham, R. François, L. Henry, and K. Müller. (2022). *Dplyr: A Grammar Data Manipulation*. [Online]. Available: https://github.com/tidyverse/dplyr

[45] J. Rodrigues Correia-Silva, R. F. Berriel, C. Badue, A. F. de Souza, and T. Oliveira-Santos, "Copycat CNN: Stealing knowledge by persuading confession with random non-labeled data," 2018, *arXiv:1806.05476*.
[46] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff Nets: Stealing functionality of black-box models," 2018, *arXiv:1812.02766*.
[47] J. H. Cheon, D. Kim, D. Kim, H. H. Lee, and K. Lee, "Numerical method for comparison on homomorphically encrypted numbers," Cryptol. ePrint Archive, Paper 2019/417, 2019. [Online]. Available: https://eprint.iacr.org/2019/417
[48] R. del Pino, S. Katsumata, T. Prest, and M. Rossi, "Raccoon: A masking-friendly signature proven in the probing model," Cryptol. ePrint Archive, Paper 2024/1291, 2024. [Online]. Available: https://eprint.iacr.org/2024/1291

BARIS BALABAN received the B.S. degree in industrial engineering from Boğaziçi University and the Ph.D. degree from the Biostatistics and Bioinformatics Department, Acıbadem Mehmet Ali Aydınlar University, under the supervision of Prof. Osman Ugur Sezerman. After working for five years with the Analytics Department, Akbank, he joined the Biostatistics and Bioinformatics Department, Acıbadem Mehmet Ali Aydınlar University, in 2019. His current research interests include privacy-preserving machine learning and machine learning applications in adult spinal deformity surgeries.

SEYMA SELCAN MAGARA received the dual degrees in electronics engineering and in computer science and engineering and the master's degree in computer science from Sabancı University, in 2018, 2019, and 2022, respectively. She is currently pursuing the Ph.D. degree with the University of Tübingen, with a focus on privacy-preserving machine learning and multi-party computation. She conducted research under the supervision of Prof. Erkay Savas.

CAGLAR YILGOR received the medical degree from Hacettepe University, Ankara, in 2007.

During his training, he was a Visiting Doctor with the University of Perugia, in 2009, where he also followed a Fellowship Program at the "Let People Move" Biomechanics Laboratory. He completed his residency from the Department of Orthopedics and Traumatology, Hacettepe University, in 2012. After finishing his Obligatory Medical Civil Service, he followed a combined Neurosurgical and Orthopedic Spinal Surgery Program at Acibadem University Maslak Hospital, Comprehensive Spine Center, from 2014 to 2016. After completing the program, he joined Dr. Ahmet Alanay's Team. He started his academic career as an Assistant Professor with the School of Medicine, Acıbadem Mehmet Ali Aydınlar University, in 2014, and became a Professor, in 2025. He is currently the Director of the Spine Fellowship Program, Comprehensive Spine Center. He is also a Spine Surgeon treating both adult and pediatric spinal disorders. He has published many scientific articles, authored several book chapters, and presented his research both nationally and internationally at scientific meetings.

Dr. Yilgor is an active member of medical associations, including the Scoliosis Research Society, the Spine Society of Europe, AO Spine, and many others. He has been an active member of the European Spine Study Group, since 2014. He is an Associate Member of the AO Spine Knowledge Forum Deformity (SKFD). He has received numerous national and international awards, including the Best Paper and Academic Achievement Awards. He is a reviewer of various scientific journals.

ALTUG YUCEKUL received the medical degree from Hacettepe University, Ankara, in 2010.

He completed his residency with the Department of Orthopedics and Traumatology, Hacettepe University, in 2017. He had one of his internships at the Cognitive Evolution Laboratory, Harvard University, USA. During his residency training, he spent a year in San Francisco, at the UCSF Spine Department, for an observership. After completing his Obligatory Medical Civil Service, he started a Spine Fellowship Program at Acibadem University Maslak Hospital, Comprehensive Spine Center. After finishing the fellowship, he joined Dr. Ahmet Alanay's Team and since then, he has been working with him as a Spine Surgeon and takes part actively in spine-related conferences and symposiums with various abstracts, posters, and presentations. He is currently a Spine Surgeon treating both adult and pediatric spinal disorders.

IBRAHIM OBEID received the degree in medicine in Lebanon, in 1998.

He was appointed as an Intern with the University Hospital Hotel Dieu de France, Beirut, Lebanon, where he became interested in spinal deformities. He graduated as an Orthopedic Surgeon, in 2003. His fellowship began first in Paris, in 2003, at the St. Joseph Hospital, and was dedicated to adult spine deformity. Since 2004, he has been with the Spine Department, Bordeaux University Hospital. In 2008, he was promoted to Staff Physician, specializing in adult and adolescent spine deformity. Since 2014, he has been with the private Clinique Du Dos, Bordeaux, Bruges. He operates 200 deformity cases and 150 degenerative spine cases per year. His practice includes pediatric and adult spinal deformity, degenerative cervical and lumbar surgery, and minimally invasive surgery but his specific areas of interest are revision and complex deformities, complex cervicothoracic spine reconstruction, and navigation. He has several publications in basic research and clinical spine pathologies, with more than 150 PubMed-referenced publications and a publication value of more than 200 impact points. He is the co-author of many textbooks on pediatric and adult spine. He participates as an Invited Faculty to many national and international meetings, especially for spinal deformity, osteotomy, and spinal alignment.

Dr. Obeid is an active member of multiple national and international spinal and spinal deformity societies, such as SFCR, SRS, and Eurospine. He is also an active member of multi-national deformity study groups leading in the spinal deformity domain (ISSG, ESSG). He is also chairing and co-chairing a multitude of spinal courses annually.

JAVIER PIZONES received the medical degree from the Medical School, Universidad Autónoma de Madrid, in 1998, and the Ph.D. degree (research thesis), in 2007.

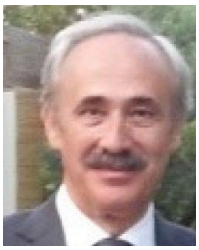
He did his Orthopedic Surgery Residency Program at Hospital Universitario de Getafe, until 2004. Since then, he has been an Attending Surgeon specializing in spine surgery. He has been the 2013 SRS-Luque Spine Fellow at Barnes and Children's Hospital St. Louis and the 2018 SRS-International Pediatric Spine Fellowship at Boston Children's Hospital. He was appointed as an Associate Orthopedic Surgery Professor with UEM University, Madrid, Spain, and has been a Faculty Member of the European Spine Diploma. He is currently a Spine Surgeon treating pediatric and adult spinal deformity at the Hospital Universitario La Paz, Madrid. He has published more than 40 peer-reviewed articles. He has more than 200 presented abstracts at national and international meetings.

Dr. Pizones joined the ESSG as an Associate Member, in 2012, and became an Executive Member, in 2019. He is a member of the Scoliosis Research Society (SRS), the Eurospine Society, and AO Spine. He has served as a Program Committee Member for GEER and Eurospine. He has been awarded the "Best Oral Presentations" in several meetings at the Spanish Spine Society (GEER), the Latin American Spine Society (SILACO), and Eurospine.

FRANK KLEINSTUECK received the medical degree from the Medical School, Johannes Gutenberg University, Mainz, and the Technical University of Munich, in 1990.

In 1998, he earned his specialty in orthopedic surgery and traumatology of the musculoskeletal system (FMH) following residency training in St. Gallen and Zürich, Switzerland. During a two-year Spine Fellowship Program at the University of California at San Francisco (UCSF) Medical Centre, from 1999 to 2001, he further subspecialized in spine surgery. He has been a Spine Surgeon with the Schulthess Clinic, Zürich, since 2004. He specializes in adult revision and adult deformity spine surgery and has been a Senior Consultant in spine surgery at the Schulthess Clinic, since 2009. Since 2013, he has been a Principal Investigator of the international multicentre "ESSG" study at the Schulthess Clinic. In addition, he lectures on spinal deformity in the study program of human motor sciences and sport with ETH Zürich, Switzerland, and supports the Outcome Assessment in Spine Surgery at the Schulthess Clinic. He is currently directing the Spine Research Program at the Schulthess Clinic. He has published more than 80 peer-reviewed articles and has presented more than 150 abstracts at national and international meetings.

Dr. Kleinstueck is a member of the Scoliosis Research Society (SRS), the Swiss Society of Spinal Surgery (SGS), and the Spine Society of Europe (Eurospine). He is a member of the Expert Committee on Spine in the Swiss Orthopedic Society.



FRANCISCO JAVIER SANCHEZ PEREZ-GRUESO received the degree from the Medical School, Universidad de Salamanca.

He did his Orthopedic Surgery Residency Program at Hospital Universitario La Paz, Madrid. After his residency, he joined the Spinal Deformity Unit, as a Staff Member starting soon after a surgical program on pediatric spinal deformity surgery. He expanded his training in different international centers specialized in spinal deformities, such as The Robert Jones and Agnes Hunt Orthopedic, Spine Disorders Department, Oswestry, U.K.; Deutsches Skoliosezentrum, Bad Wildungen, Germany; Hospital Saint Vincent de Paul, Paris; and Hospital for Special Surgery, New York, NY, USA. He also has been involved in outreach programs in Ghana, West Africa, from 2003 to 2014. He was promoted to the Chief of the Spine Unit Hospital La Paz, in 2006 until his retirement in October 2018. He has been appointed as an Emerito del Servicio Madrileño de Salud developing his research activity at Hospital Universitario La Paz. He is currently a Spine Surgeon specializing in pediatric and adult spinal deformity. He has published more than 50 peer-reviewed articles and more than 100 podium presentations in national and international meetings.

Dr. Perez-Grueso was an active member of different spine societies becoming the President of the Spanish Spine Society (GEER), from 2004 to 2006, and the Director at Large of the Board of Directors, Scoliosis Research Society, from 2010 to 2012. He joined ESSG as an Executive Member, in 2011.

FERRAN PELLISÉ received the Ph.D. degree (cum laude), in 1994.

He was trained as an Orthopedic Surgeon with the Vall d'Hebron University Hospital, Barcelona. Since 1995, he has been entirely dedicated to the surgical treatment of spinal disorders with a special interest in scoliosis and other spinal deformities. In 2010, he founded the European Spine Study Group (ESSG) together with other well-known researchers in the field of adult spinal deformities. Today, ESSG is the main platform for the analysis and evaluation of adult spinal deformities in over Europe. He is currently the Chief of the Spine Unit, Hospital Universitari Vall d'Hebron; the Director of the Spine Service at Hospital Quirón Barcelona; and an Associate Professor with the Universitat Autónoma de Barcelona.

Dr. Pellisé is a member of the boards of several academic journals. He has been a member of the boards of several medical associations, including the Spanish Spine Society (President), the Spine Society of Europe (President), and the Scoliosis Research Society (Director at Large). From 2005 to 2015, he was the Deputy Editor of the *European Spine Journal*.

AHMET ALANAY received the degree from the Faculty of Medicine, Ankara University, in 1988.

He completed his residency with the Department of Orthopedics and Traumatology, Hacettepe University, from 1991 to 1996. He started his career as an Assistant Professor with the Department of Orthopedics and Traumatology, Hacettepe University, where he became an Associate Professor, in 2002, and a Professor, in 2007. He also completed a fellowship program on spinal deformities with the University of Kansas Medical Center and was a Visiting Professor with the UCLA School of Medicine, from 2005 to 2006. He established the Comprehensive Spine Center, Acibadem Maslak Hospital, in 2013. Since then, he has been the Medical Director of the Comprehensive Spine Center. He is currently a Spine Surgeon treating both adult and pediatric spinal disorders. He is also a Faculty Member with the Acibadem University School of Medicine. Each year, with his team, he treats over 2500 outpatients, operates approximately 200 patients, and publishes more than 20 peer-reviewed publications in high-ranking international journals along with numerous national and international presentations at scientific meetings.

Dr. Alanay is an active member of medical associations, including the American Academy of Orthopedic Surgeons, the Scoliosis Research Society, the North American Spine Society, the Spine Society of Europe, and many others. From 2014 to 2016, he was a member of the Board of Directors of Scoliosis Research Society, where he still serves on the Education Committee and Safety and Value Committee. From 2006 to 2012, he was also on the executive committee of the Spine Society of Europe. He has been a Founding Member of the European Spine Study Group. He has been the pioneer of vertebral body tethering surgery in Europe and has received many national and international honors and awards for his scientific studies. He contributes to the training of surgeons through the courses he organizes all around the world.

ERKAY SAVAS (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Electronics and Communications Engineering Department, Istanbul Technical University, in 1990 and 1994, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering (ECE), Oregon State University, in June 2000. He had worked for various companies and research institutions before he joined Sabancı University, in 2002. He has been the Dean of the Faculty of Engineering and Natural Science, Sabancı University, since July 2020. His research interests include applied cryptography, data and communication security, privacy in biometrics, security and privacy in data mining applications, embedded systems security, and distributed systems. He is a member of ACM, the IEEE Computer Society, and the International Association of Cryptologic Research (IACR).

ÇETIN BAĞCı received the degree from the Department of Computer Engineering, Ege University, in 1986. As a founding partner of Bilmed Computer and Software Inc., he continues to serve as the Deputy General Manager responsible for software and research and development. He has experience managing research and development projects, innovation processes, and technology development. He has worked for many years as a Software Developer, particularly in health and supplementary health insurance, and continues his career as a Manager.

OSMAN UGUR SEZERMAN received the B.Sc. degree from the Electrical Engineering Department, Bogaziçi University, Istanbul, Türkiye, the M.Sc. degree from the Biomedical Engineering Department, Bogaziçi University, and the Ph.D. degree in biomedical engineering from Boston University, Boston, MA, USA.

He was with Boston University and Boğaziçi University as a Researcher and an Instructor. From 1999 to 2015, he was with the Biological Sciences and Bioengineering Program, Sabancı University, where he established the Computational Biology and Protein Engineering Laboratory. He is currently the Head of the Biostatistics and Medical Informatics Department, Acıbadem Mehmet Ali Aydınlar University, Istanbul. His current research interests include personalized and precision medicine, protein engineering, drug and vaccine design, functional genomics, metagenomics, systems biology, and bioinformatics.