applied sciences

MDPI

Article

Framework for Privacy-Preserving Wearable Health Data Analysis: Proof-of-Concept Study for Atrial Fibrillation Detection

Anamaria Vizitiu [1,2,*], Cosmin-Ioan Nita [1,2], Radu Miron Toev [2], Tudor Suditu [2], Constantin Suciu [1,2] and Lucian Mihai Itu [1,2]

[1] Department of Automation and Information Technology, Transilvania University of Braşov, 500174 Braşov, Romania; cosmin.nita@siemens.com (C.-I.N.); constantin.suciu@siemens.com (C.S.); lucian.itu@siemens.com (L.M.I.)
[2] Advanta, Siemens SRL, 500097 Braşov, Romania; radu.toev@siemens.com (R.M.T.); tudor.suditu@siemens.com (T.S.)
* Correspondence: anamaria.vizitiu@unitbv.ro

Abstract: Medical wearable devices monitor health data and, coupled with data analytics, cloud computing, and artificial intelligence (AI), enable early detection of disease. Privacy issues arise when personal health information is sent or processed outside the device. We propose a framework that ensures the privacy and integrity of personal medical data while performing AI-based homomorphically encrypted data analytics in the cloud. The main contributions are: (i) a privacy-preserving cloud-based machine learning framework for wearable devices, (ii) CipherML—a library for fast implementation and deployment of deep learning-based solutions on homomorphically encrypted data, and (iii) a proof-of-concept study for atrial fibrillation (AF) detection from electrocardiograms recorded on a wearable device. In the context of AF detection, two approaches are considered: a multi-layer perceptron (MLP) which receives as input the ECG features computed and encrypted on the wearable device, and an end-to-end deep convolutional neural network (1D-CNN), which receives as input the encrypted raw ECG data. The CNN model achieves a lower mean F1-score than the hand-crafted feature-based model. This illustrates the benefit of hand-crafted features over deep convolutional neural networks, especially in a setting with a small training data. Compared to state-of-the-art results, the two privacy-preserving approaches lead, with reasonable computational overhead, to slightly lower, but still similar results: the small performance drop is caused by limitations related to the use of homomorphically encrypted data instead of plaintext data. The findings highlight the potential of the proposed framework to enhance the functionality of wearables through privacy-preserving AI, by providing, within a reasonable amount of time, results equivalent to those achieved without privacy enhancing mechanisms. While the chosen homomorphic encryption scheme prioritizes performance and utility, certain security shortcomings remain open for future development.

Keywords: privacy preservation; homomorphic encryption; electrocardiogram; deep learning; cloud framework; wearable sensor data; atrial fibrillation

Citation: Vizitiu, A.; Nita, C.-I.; Toev, R.M.; Suditu, T.; Suciu, C.; Itu, L.M. Framework for Privacy-Preserving Wearable Health Data Analysis: Proof-of-Concept Study for Atrial Fibrillation Detection. *Appl. Sci.* **2021**, *11*, 9049. https://doi.org/10.3390/app11199049

Academic Editor: Paolo Visconti

Received: 12 August 2021
Accepted: 26 September 2021
Published: 28 September 2021

1. Introduction

Wearable medical devices provide remote solutions for patients to monitor their well-being and vital signs. These devices are designed to capture and collect various types of health data and, in recent years, coupled with data analytics, cloud computing, and artificial intelligence, even allow for early disease detection [1]. Many popular developments are related to the cardiology field, e.g., monitoring of the heart rate, and of the electrical activity of the heart through electrocardiograms (ECG).

Such heart monitors are useful in detecting signs of atrial fibrillation, the most common type of cardiac arrhythmia. Atrial fibrillation causes irregular and abnormally fast

heartbeats, representing the leading cause of stroke [2]. Early and precise identification of atrial fibrillation is therefore an active field of research aimed at improving the prevention of heart failure. Although an ECG can be used to detect irregularities in the heartbeat, atrial fibrillation is more difficult to trace because of the episodic nature of the symptoms.

Wearable ECG devices can help track and record irregular heartbeats by facilitating low-cost ECG monitoring. Although not entirely a wearable, AliveCor's portable single-lead personal ECG device is one of the most widely used heart rate monitors. It has been FDA-cleared for the detection of atrial fibrillation and is currently widely used in research studies and clinical practice [3].

To detect atrial fibrillation using wearable devices, such as smartwatches or portable ECG devices, the acquired data are automatically sent to a cloud service and analyzed for atrial fibrillation using algorithms powered by artificial intelligence (AI). Within seconds, the smartphone application displays the results of the analysis. The information collected from several users may be stored in a centralized storage system and used to continuously improve AI-based detection systems. When personal health information is stored, shared, and manipulated, there is always an inherent risk of privacy breaches, and, hence, privacy and security issues arise.

In recent years, considerable work has been expended on the creation of various privacy-preserving approaches capable of addressing the gap between data privacy and utility, as demanded by the recent focus on privacy-related aspects. Among these techniques, homomorphic encryption has made rapid progress. While sufficiently secure, many homomorphic encryption-based techniques exhibit suboptimal performance due to their high computational complexity, computations being several orders of magnitude slower than on plaintext. This clearly restrains their usability in real-world applications.

To prevent any issues while employing AI to enhance the functionality of wearables and improve patient care, technical solutions have to simultaneously meet privacy and utility requirements [4]. Although many promising studies have been recently published on the use of homomorphic encryption in machine learning workflows, homomorphic encryption has not been broadly adopted in crowd-sensing systems. In this work, we address privacy issues related to wearable devices and propose a framework to ensure the privacy and integrity of personal medical data, with emphasis on performance analysis and demonstration in cardiac arrhythmia detection. We aim to address data privacy concerns in a way that enables the sharing of patient-related health information while ensuring the utility of AI in atrial fibrillation detection on real-world data. The paper focuses on both the training and deployment of AI-based atrial fibrillation models directly on homomorphically encrypted data without sharing or storing users' personal data in the cloud.

The contributions of this work can be summarized as follows: (i) a privacy-preserving cloud-based machine learning framework for wearable devices, (ii) CipherML—a library for fast implementation and deployment of deep learning-based solutions on homomorphically encrypted data, and (iii) a proof-of-concept study for atrial fibrillation detection using electrocardiograms recorded on a wearable device, where a comparison against state-of-the-art results is performed to evaluate the performance loss caused using homomorphically encrypted data. To the best of our knowledge, this is the first cloud-based framework that integrates deep learning, and homomorphic encryption as a privacy-preserving mechanism for training and deploying personalized medicine solutions using sensitive medical data collected by on-device sensors.

The remainder of this paper is organized as follows. Section 2 descriebes the related work. Section 3 presents the methods. Section 4 describes the results in terms of classification performance and runtime. Finally, conclusions are drawn and the remaining issues are presented in Sections 5 and 6.

2. Related Work

2.1. Atrial Fibrillation Detection

Machine learning algorithms have the potential to improve heart abnormality detection and classification by automatically learning and extracting atrial fibrillation patterns from large volumes of ECG data.

Many atrial fibrillation detection methods strongly rely on strategies that convert ECG recordings into a representation suitable for distinguishing between normal and abnormal heartbeats. Consequently, methods based on time, frequency, and morphological features derived from ECG recordings have been extensively implemented to automatically detect and identify episodes of atrial fibrillation [5]. Classifiers used to detect atrial fibrillation from hand-crafted features range from Support Vector Machine (SVM), decision trees, Multi-Layer Perceptron (MLP) to more advanced models constructed as a cascade or combination of different models. However, the efficiency of detection depends on the consistency and utility of the hand-crafted features.

Deep learning-based classifiers have achieved remarkable state-of-the-art results in a wide range of domains, including healthcare. Consequently, solutions for detecting cardiac arrhythmia in electrocardiograms based solely on the raw signal, with no feature engineering, are becoming more prevalent [6]. Unlike traditional machine learning algorithms, deep learning algorithms have feature engineering modules embedded into the model. When the model is exposed to large volumes of data during the training phase, the relevant features are extracted in line with the problem domain. Essentially, deep learning models learn to derive an abstract representation from the ECG signals and, at the same time, categorize the recording based on the extracted features, whereas traditional machine models learn how to categorize ECG signals based on hand-crafted features.

Despite clear challenges that arise in designing and selecting appropriate features from ECG signals, feature-based machine learning models trained on hand-crafted features are still widely applied for atrial fibrillation detection. This tendency is observed in several works that explored the ability of machine learning and deep learning techniques to enhance early detection of atrial fibrillation on short recordings acquired using AliveCor's single-channel ECG device [7]. Overall, approaches vary from hand-crafted features fed to a Random Forest (RF) or MLP to direct processing of the raw signal by Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNNs). However, the results revealed that models based on features extracted from ECG recordings performed slightly better in differentiating atrial fibrillation from noise, normal or other rhythms in short-term ECG recordings.

2.2. Privacy Preservation

Current solutions for privacy preservation in medical wearables use ordinary encryption technologies (e.g., symmetric-key encryption, public-key encryption) to protect sensitive patient well-being data only when the data are stored on the server or in transit. Any manipulation of the data at rest requires the data to be first decrypted. This, however, leads to sensitive information leakage issues. A promising solution is based on homomorphic encryption. This specific form of encryption allows data to be encrypted while it is being manipulated. The data can remain private on the server by enabling data processing without revealing the underlying information. Since the mathematical properties of the data are preserved in the encrypted form, homomorphic encryption becomes a viable option to guarantee anonymity while providing sufficient utility.

While homomorphic encryption represents a promising high-level option for privacy-preserving computations, certain practical challenges remain to be addressed, including enhanced security, shorter runtime, and improved accuracy. A variety of new homomorphic encryption schemes with different capabilities and constraints have recently appeared in the field of homomorphic encryption [8,9]. Most of these schemes have significant practical limitations due to their high computational complexity, lack of direct support for floating-point data, and the constraint of performing a limited number of operations

on homomorphically encrypted data to avoid losing the underlying information through noise accumulation.

The first important work that tackles the idea of privacy-preserving neural networks through homomorphic encryption was suggested by Orlandi et al. [10] in 2007. The encryption scheme considered therein allowed only a few basic operations to be performed in the encrypted domain. Hence, interactions between the data owner and the server were required for the remaining operations. CryptoNets [11] eliminates the interaction between the involved parties using a Fully Homomorphic Encryption (FHE) scheme in combination with the polynomial approximation of non-linear activation functions. A performance limitation that restricts usability is encountered when handling large networks, due to the computational complexity of FHE. Chabanne et al. [12] have further enhanced the capabilities of CryptoNets and introduced a low-degree polynomial approximation for the ReLU function which, in addition to the batch normalization technique, has been shown to reduce the depth limits of FHE to some extent. Focusing on improving the efficiency of neural networks when operating on encrypted data, CryptoDL [13] proposed to approximate sigmoid and tanh functions with low-degree polynomials. GAZELLE [14] suggested a different approach by combining additive homomorphic encryption with garbled circuit techniques (e.g., functions expressed as circuits of logical gates). None of these techniques, however, address the issue of privacy-preserving training for deep neural network models.

Most, if not all, standard encryption schemes are based on modular arithmetic, and their security relies on the difficulty of factoring large numbers. This makes standard encryption schemes exceptionally secure. The key downside of most of the privacy-preserving neural network solutions is the computational overhead: deeper networks need more computations, resulting in longer running times. In addition, an encoding strategy for floating point numbers has been typically used to empower computations on real-world data. Not only does the encoding strategy explicitly restrict the utility of these methods, but it also has a direct impact on the outcome of the computations. The attempt to address the non-linearity property in neural network models through an approximation mechanism does not necessarily result in better performance. In fact, most of these homomorphic solutions fail to maintain the highest prediction accuracy due to the polynomially approximated activation functions. As a consequence, without substantial development, practical adoption is bounded.

Alternatively, with a focus on enabling privacy-preserving deep learning-based analysis in real-world applications, Vizitiu et al. [15] proposed an approach in which a simplified encryption scheme based on linear transformations is adapted for both training and inference on encrypted data. The approach is based on linear transformation and relies on the non-commutativity property of matrix algebra. Because of this, the encryption scheme has practical advantages as it requires significantly less computational cost, is noise-free (allowing for an unlimited number of operations without noise accumulation) and implicitly allows for computations on real numbers. It is, however, associated with weaker security compared to standard homomorphic encryption schemes.

Numerous open-source homomorphic encryption libraries have arisen in recent years to facilitate the development and use of homomorphic encryption for AI models, including SEAL, HElib, TFHE, PALISADE, cuHE, HEAAN, and HE-transformer [16]. Microsoft's Simple Encrypted Arithmetic Library (SEAL) [17], with support for the Brakerski/Fan-Vercauteren (BFV) [18] and Cheon-Kim-Kim-Song (CKKS) [19] schemes, and IBM's HELib [20] based on the Brakerski-Gentry-Vaikuntanathan (BGV) [21] scheme, are two of the most widely used libraries for building privacy-preserving neural network inference applications. The first and most obvious shortcoming of HELib and SEAL is their noise accumlation behaviour during homomorphic operations, which severely limits the operations that may be done on ciphertexts. Noise-management mechanisms are used to maintain the noise level below a predetermined threshold, hence preventing the ciphertext from becoming corrupted. To accommodate for this, the calculation depth must be specified in advance to provide an adequate noise budget and assure proper decryption.

The "bootstrapping" technique, which allows for the reduction of noise in ciphertexts when it exceeds a predefined threshold, is regarded inefficient and is frequently the bottleneck of privacy-preserving AI models. For example, inference using a ResNet-20 model on the CIFAR-10 dataset takes approximately four hours when employing the SEAL and CKKS techniques [22]. While there are some works that significantly improve bootstrapping time [23,24], they consider ciphertexts as single-bit encryptions and use logic gate operations to design arbitrary algorithms as circuits. Additionally, neither HElib nor SEAL directly support floating-point numbers. They treat ciphertexts as integers (or approximations thereof) and perform only multiplication and addition on them. As a result, implicit division is not supported, and non-linear functions must be approximated by low-degree polynomials.

Three objectives are critical in the development of privacy-preserving AI models: accuracy, data security, and computational complexity [16]. Extremely safe models are ineffective if their accuracy and computing complexity are inadequate. As a result, depending on the nature of the problem, a trade-off must be made when selecting the encryption mechanisms.

3. Materials and Methods

3.1. Privacy-Preservation with Homomorphic Encryption

To address potential privacy issues in wearable devices, we propose a cloud-based framework that relies on homomorphic encryption as a mechanism to guarantee the privacy and integrity of a person's personal health information when wearable devices are used for remote health monitoring. Encryption takes place on the wearable device, and the encrypted form of the collected information (ciphertext) is sent to a cloud service for processing and/or storage. Data processing via AI-based systems can be performed in the cloud computing environment directly on homomorphically encrypted data without the need to decrypt the data or access the decryption key.

Inspired by the work of Vizitiu et al. [15], a variant of the MORE (Matrix Operation for Randomization or Encryption) encryption scheme [25] is used to perform privacy-preserving computations within neural network models.

The scheme encrypts plaintexts $m_1 \in \mathbb{R}$ and $m_2 \in \mathbb{R}$ in ciphertexts $\mathbf{C_1} = \mathbf{SM_1S^{-1}}$ and $\mathbf{C_2} = \mathbf{SM_2S^{-1}}$, where $\mathbf{M_1} = \begin{pmatrix} m_1 & 0 \\ 0 & r_1 \end{pmatrix}$ and $\mathbf{M_2} = \begin{pmatrix} m_2 & 0 \\ 0 & r_2 \end{pmatrix}$ is constructed using a random parameter $r_1 \in \mathbb{R}$ and $r_2 \in \mathbb{R}$, respectively, and $\mathbf{S} \in \mathbb{R}^{2\times 2}$ defines the symmetric secret key. Here the matrix-based secret key $\mathbf{S}$ is chosen to be invertible.

Decryption is done by computing $\mathbf{K_1} = \mathbf{S^{-1}C_1S}$ and $\mathbf{K_2} = \mathbf{S^{-1}C_2S}$. Finally, the plaintexts m_1 and m_2 is extracted as $m_1 \leftarrow \mathbf{K_1}_{(1,1)}$ and $m_2 \leftarrow \mathbf{K_2}_{(1,1)}$, respectively.

The homomorphic properties of this scheme can be identified easily using linear algebra. Encryption of $m_1 + m_2$ is computed as

$$\mathbf{C_1} + \mathbf{C_2} = \mathbf{SM_1S^{-1}} + \mathbf{SM_2S^{-1}} = \mathbf{S}(\mathbf{M_1} + \mathbf{M_2})\mathbf{S^{-1}} \tag{1}$$

and $m_1 \times m_2$ is computed as

$$\mathbf{C_1} \times \mathbf{C_2} = \mathbf{SM_1S^{-1}SM_2S^{-1}} = \mathbf{S}(\mathbf{M_1M_2})\mathbf{S^{-1}}. \tag{2}$$

Similarly, it can be shown that this is also true for subtraction and division. Moreover, non-linear functions that can be evaluated directly on a matrix (e.g., logarithm, exponentiation, square root) can also be evaluated directly on a ciphertext. Other non-linear functions can still be constructed by approximating them with a finite polynomial series, leading to a sequence of elementary algebraic operations. This approach allows one to perform an arbitrary number of operations, with arbitrary complexity, without loss of precision. This is the main feature that enables the implementation of complex learning-based algorithms on encrypted data.

MORE is a symmetric-key algorithm where a key generated according to Algoritm 1 is used for both encryption (Algorithm 2) and decryption (Algorithm 3).

Algorithm 1 MORE secret key generation

Output: Secret key $\boldsymbol{S} \in \mathbb{R}^{2\times 2}$
function KEYGENERATION()
  while True **do**
    $\boldsymbol{S} \leftarrow \text{RandomUniform}(size = (2,2), minval, maxval)$
    if $\det(\boldsymbol{S}) \neq 0$ **then** // Ensure matrix invertibility
      break
    end if
  end while
  return $\boldsymbol{S}$
end function

Algorithm 2 MORE encryption

Input: Plaintext data $m \in \mathbb{R}$
Input: Secret key $\boldsymbol{S} \in \mathbb{R}^{2\times 2}$
Output: Ciphertext $\boldsymbol{C} \in \mathbb{R}^{2\times 2}$
function ENCRYPTION($m, \boldsymbol{S}$)
  $\boldsymbol{M} \in \mathbb{R}^{2\times 2} \leftarrow$ zero matrix
  $\boldsymbol{M}(\mathbf{0},\mathbf{0}) \leftarrow m$
  $\boldsymbol{M}(\mathbf{1},\mathbf{1}) \leftarrow \text{RandomUniform}(minval, maxval)$
  $\boldsymbol{C} \leftarrow \boldsymbol{S} \times \boldsymbol{M} \times \boldsymbol{S}^{-1}$
  return $\boldsymbol{C}$
end function

Algorithm 3 MORE decryption

Input: Ciphertext $\boldsymbol{C} \in \mathbb{R}^{2\times 2}$
Input: Secret key $\boldsymbol{S} \in \mathbb{R}^{2\times 2}$
Output: Plaintext data $m \in \mathbb{R}$
function DECRYPTION($\boldsymbol{C}, \boldsymbol{S}$)
  $\boldsymbol{K} \leftarrow \boldsymbol{S}^{-1} \times \boldsymbol{C} \times \boldsymbol{S}$
  $m \leftarrow \boldsymbol{K}(\mathbf{0},\mathbf{0})$
  return m
end function

3.2. CipherML

The proposed framework uses CipherML (Siemens SRL, Brasov, Romania)—an in-house developed library for privacy-preserving machine learning, which is based on the MORE encryption scheme. The library can be used to train standard neural network models, but also to make inferences (predictions) based on MORE homomorphically encrypted data.

Neural network models are composed of repetitive blocks of computation based on a small set of simple operations over rational numbers (e.g., multiplication, addition, division, subtraction, exponential and logarithm). By leveraging the MORE scheme's homomorphism and direct support for floating point data, the functionality of neural network models can be formulated to account for operations on ciphertext data (e.g., scalar multiplication becomes matrix multiplication), enabling cloud-based training and inference on homomorphically encrypted data in a manner similar to that of classical neural networks (i.e., standard feed-forward and backpropagation). Hence, CipherML extends the functionality of classical neural network models to be available for operations on MORE homomorphically encrypted data.

The library provides a high-level Keras-like API [26] that enables quick prototyping of privacy-preserving neural network models with minimal code. The learning-based models can be defined by stacking one building block on top of the other, and then trained

on MORE homomorphically encrypted data by simply calling the fit method. Written in the C++ programming language, the CipherML library includes implementations of widely used building blocks in neural networks, such as computational layers, activation functions, objective functions, and optimizers. In addition, the library provides support for convolutional neural networks through temporal and 2D spatial convolution layers. It also includes other common utility layers, such as dropout and average pooling for both 1D temporal data and 2D spatial data. Max pooling cannot be formulated because it is not possible to perform comparison operations on ciphertext data. As for activation functions, the current version of CipherML includes the sigmoid, hyperbolic tangent (tanh), and softmax functions. These non-linear functions are formulated as matrix functions. Stochastic Gradient Descent (SGD) is currently the only optimizer available. Other optimizers (e.g., ADAM [27]) may be integrated to improve performance. More advanced training strategies such as gradient clipping, learning rate decay, or early stopping are not supported because they rely on comparison operations (ciphertext values cannot be used in conditional statements).

3.3. Atrial Fibrillation Detection

In this section, we present the application of the proposed framework for detecting atrial fibrillation from short single-lead ECG recordings. Due to the existence of alternate rhythms that exhibit heartbeat patterns similar to atrial fibrillation, atrial fibrillation may be misdiagnosed when based on short-term detection. We propose a privacy-preserving solution to distinguish atrial fibrillation from normal rhythms, other abnormal rhythms, and noisy ECG recordings.

In the current methodology, atrial fibrillation detection models are trained offline using publicly available datasets. Given the proposed framework and the features of the MORE encryption scheme, there are two possible training scenarios. Specifically, AI models can be trained offline on unencrypted data and then deployed to the cloud for inference on encrypted data, or they can be trained directly on encrypted data. The second approach is particularly useful in real-world scenarios in which confidential health-related training datasets are not made publicly available to third parties (i.e., they do not leave the data provider). Using the proposed framework, the training dataset is encrypted at the data provider (e.g., hospital), and only the encrypted version is transferred and stored on the cloud for privacy-preserving AI model training. Moreover, this approach favors on-line training, in which AI models are retrained and fine-tuned as new data becomes available.

Following the strategies used in the research community to detect cardiac arrhythmias, we consider two approaches: (i) a classical machine learning solution based on feature engineering, and (ii) a deep learning end-to-end approach where the detection is based on the raw ECG signal. To address the problem of confidentiality of personal data, homomorphically encrypted data are used for both training and prediction. The proposed deep learning-based ciphertext data analysis is presented in Algorithm 4 where all operations performed during training and prediction are formulated in accordance with Section 3.1. In the following subsection, we describe the dataset, the experimental setup, including the proposed models, and their performance.

3.3.1. Dataset

ECG recordings provided for the Physionet 2017 Challenge [28] have been employed herein. The ECG dataset contains recordings obtained with AliveCor's single-channel ECG device and includes a total of 8528 recordings, each labeled as normal, atrial fibrillation, other abnormal rhythms, or noisy ECG samples. Of the total 8528 ECG recordings, 5076 recordings are labeled as "Normal", 758 recordings are labeled as "Atrial Fibrillation", 279 recordings are labeled as "Noise", and 2415 recordings are labeled as "Other". Examples of ECG recordings for each category are shown in Figure 1.

Algorithm 4 Deep learning-based analysis on ciphertext data

function TRAINONCIPHERTEXT()
    $\boldsymbol{X}_{train}, \boldsymbol{Y}_{train} \leftarrow$ LoadDataset()
    $\boldsymbol{X}_{train} \leftarrow$ Preprocess($\boldsymbol{X}_{train}$)
    $\boldsymbol{S} \leftarrow$ KeyGeneration()
    $\boldsymbol{X}_{train_{enc}} \leftarrow$ Encryption($\boldsymbol{X}_{train}, \boldsymbol{S}$)
    $\boldsymbol{Y}_{train_{enc}} \leftarrow$ Encryption($\boldsymbol{Y}_{train}, \boldsymbol{S}$)
    BuildModel()
    Train($\boldsymbol{X}_{train_{enc}}, \boldsymbol{Y}_{train_{enc}}$)
    return $model_{enc}$
end function

function PREDICTONCIPHERTEXT()
    $\boldsymbol{X}_{test} \leftarrow$ LoadSamples()
    $\boldsymbol{X}_{test} \leftarrow$ Preprocess($\boldsymbol{X}_{test}$)
    $\boldsymbol{S} \leftarrow$ LoadKey()
    $\boldsymbol{X}_{test_{enc}} \leftarrow$ Encryption($\boldsymbol{X}_{test}, \boldsymbol{S}$)
    LoadModel()
    $\tilde{\boldsymbol{Y}}_{test_{enc}} \leftarrow$ Predict($\boldsymbol{X}_{test_{enc}}$)
    $\tilde{\boldsymbol{Y}}_{test} \leftarrow$ Decryption($\tilde{\boldsymbol{Y}}_{test_{enc}}, \boldsymbol{S}$)
    return $\tilde{\boldsymbol{Y}}_{test}$
end function

3.3.2. Feature-Based Method

The proposed privacy-preserving machine learning-based method is inspired by the solution of Datta et al. [29], which secured the top position in the Physionet 2017 Challenge. Atrial fibrillation is identified by first extracting relevant features from the ECG and then feeding the resulting features to a classifier. To deal with the largely imbalanced training dataset that could lead to poor detection performance, Datta et al. [29], chose to classify ECG recordings using a cascade of binary classifiers.

The first module classifies an ECG signal into one of two intermediate classes ("Normal" + "Others" and "AF" + "Noisy"). Depending on the outcome of the first classifier, target-specific features are derived from the recording and fed to the corresponding binary classifier (i.e., "Normal" vs. "Others" and "AF" vs. "Noisy") for the final classification. Hence, the first binary classifier is responsible for identifying the intermediate class, while the two binary classifiers from the second stage of the pipeline have to recognize the specific type of rhythm. In each of the three binary classifiers, a distinct set of features is used. In a privacy-preserving scenario, such an approach would require access to the intermediate detection results. To avoid the disclosure of intermediate classification results, we have opted for an end-to-end classifier based on neural network. Hence, a single multi-class classifier is employed instead of a multi-layer cascaded binary classifier, and all features are used regardless of their type. The final input feature set is obtained by combining the three feature sets considered in [29].

The proposed computer-aided detection system consists of three main components: noise removal, feature extraction, and classification. The feature extraction step relies heavily on the characteristics of the data and cannot be performed in the encrypted format. Therefore, the data preprocessing step must be performed on the device. The proposed workflow is outlined in Figure 2. Before the ECG data are analyzed by the machine learning algorithm, it is preprocessed on the device. The preprocessing step includes noise removal, feature engineering, and data normalization. The resulting information is then homomorphically encrypted with a secret key. Thereafter, the cloud-based machine learning classifier receives only the encrypted form of the data (ciphertext), while the raw data (plaintext) remains private on the device (at the user). Finally, since CipherML provides direct support for floating-point arithmetic and all operations performed within the neural network are formulated by exploiting the homomorphic properties of MORE,

the network can be further used for detection. This network produces encrypted results that can only be interpreted after decryption with the secret key.

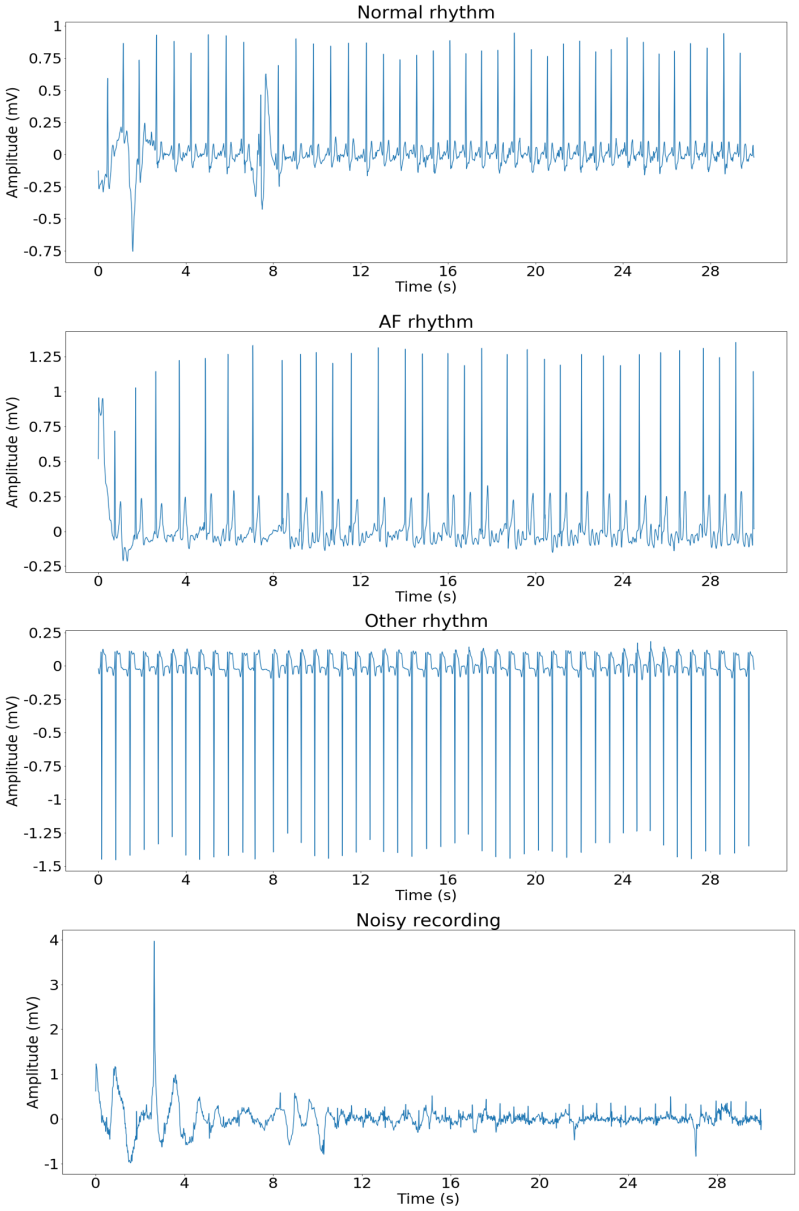

Figure 1. Examples of different classes of ECG recordings.

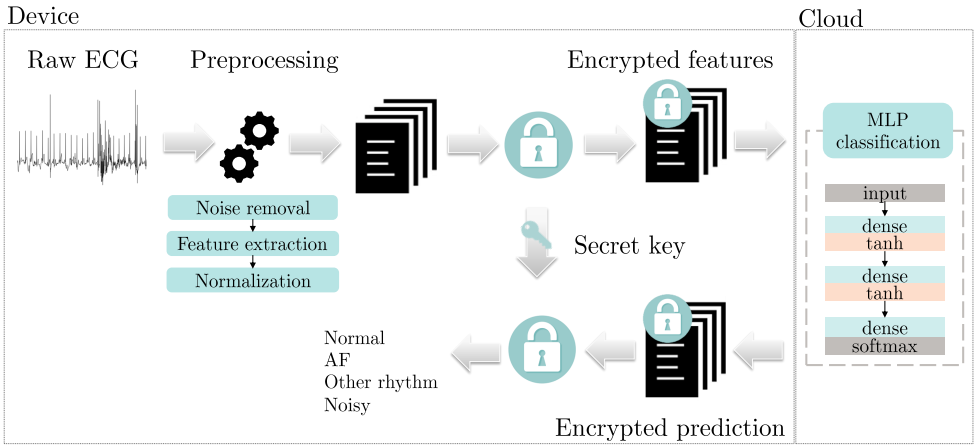

Figure 2. Workflow of the proposed privacy-preserving feature-based ECG classification method.

Feature Engineering

The waveform of a normal sinus rhythm is represented on the ECG by three distinct waves: the P-wave, the QRS-complex, and the T-wave. However, in atrial fibrillation, chaotic fibrillatory waves may influence the morphology of the P-wave [5]. Moreover, the time interval from R peak to R peak is known to vary between beats. Consequently, cardiac arrhythmias are typically identified by analyzing the morphological and rhythmic properties of the electrocardiogram [30]. A total of 188 features were selected, including morphological features derived from the P, Q, R, S, and T points, features related to heart rate variability, frequency domain features, and statistical features based on RR intervals. A more detailed discussion of the feature engineering part is provided in [29]. To lessen the impact of noise in the feature extraction process, the ECG signal recorded by the wearable device is first denoised using a combination of a spectrogram-based technique and high pass filtering [29]. Since the extracted features have different ranges of values, the value of each feature is rescaled to have zero-mean and unit variance. The resulting features are homomorphically encrypted using the MORE encryption scheme before being sent to the classifier.

MLP Classifier

As depicted in Figure 2, a multi-layer perceptron is proposed as a classifier. This model takes as input 188 homomorphically encrypted features and consists of two fully connected hidden layers, each followed by a tanh activation function. The first hidden layer has 100 units while the second has 10 units. Four units are considered in the fully connected output layer and a softmax activation function is applied to classify an ECG recording based on encrypted features.

The model uses the following hyperparameters for training: a learning rate of 0.01, a batch size of 128, 100 epochs, and SGD as the optimizer. The objective function is the categorical cross-entropy.

3.3.3. End-to-End Method

To mitigate the need for exhaustive feature extraction on the wearable device, we also propose a deep convolutional neural network model that enables feature learning directly from the short-term ECG recordings. In this workflow, the ECG signal recorded by the device is homomorphically encrypted on the device and sent to the cloud for analysis and classification. To speed up the computations in deep-learning models, the data are normalized to zero mean and unit variance. Moreover, since the length of ECG recordings in the database ranges from 9 to 61s, and to maximize the detection accuracy for atrial fibrillation, the maximum length is considered for analysis. Therefore, ECG recordings shorter than 61 s are zero-padded. No further transformations (e.g., denoising) are per-

formed in the preprocessing part (data padding and normalization are performed on the device). However, these operations can also be performed on the cloud.

The proposed workflow is outlined in Figure 3. The deep learning-based classifier has two main components: the CNN-based feature extractor and the MLP-based classifier. The network takes as input a 61-second encrypted ECG recording with a sampling rate of 300 Hz. Hence, the encrypted ECG signal goes through three stages of convolution, non-linearity, and down-sampling, through which relevant features are extracted. One-dimensional convolutional layers are used to learn local features (i.e., feature maps) from the time sequence data, while average pooling layers are used to reduce the dimension of the data. By stacking a series of such operations, the network is able to provide a hierarchy of features. The initial feature maps are obtained by convolving the encrypted ECG signal with 32 filters of kernel size 55×1. The resulting feature maps are passed to a tanh non-linearity function, and their complexity is reduced by a factor of 10 through the pooling operation. The down-sampled feature maps are further used as input to the next block of computation, which generates a set of 32 new feature maps by convolving the input feature maps with 32 filters of kernel size 25×1. Again, their resolution is reduced by a factor of 5 by the pooling operation. The last convolution block uses 32 filters of kernel size 10×1, tanh activation, and a pooling size of 5. The feature extraction process ends when the data reaches the classifier. Consequently, the resulting features are fed into an MLP-based network that classifies the ECG recording into one of the four classes. The MLP-based classifier consists of a fully connected hidden layer with 32 units and tanh activation function, followed by a fully connected output layer with four units and a softmax function.

The encrypted parameters of the network are optimized over 100 epochs to minimize categorical cross-entropy loss. Training is performed with batches of 32 data samples using SGD and a learning rate of 0.01.

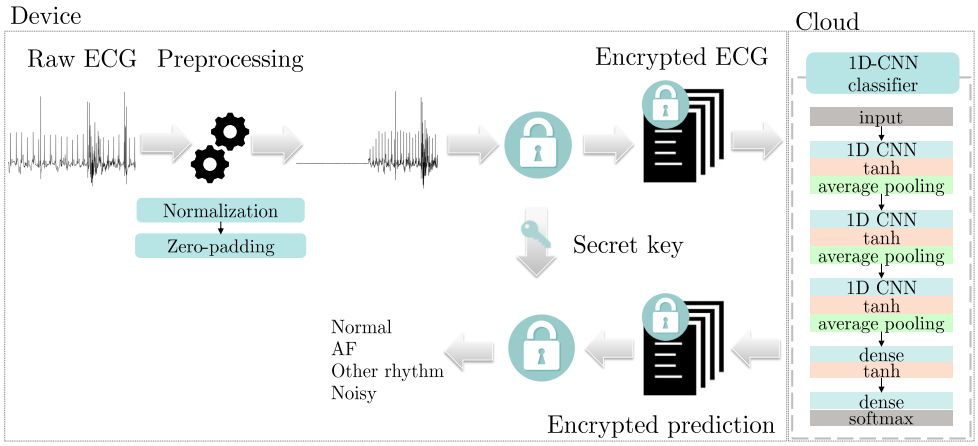

Figure 3. Workflow of the proposed privacy-preserving end-to-end ECG classification method.

3.4. Cloud Implementation Considerations

In this section, we highlight important cloud implementation considerations. Our implementation is based on Google's Cloud Platform [31], but in principle, any other public or private cloud platform can be employed instead. We chose Google's Cloud Platform because they currently offer the most mature AI services [32].

The platform uses event-driven services. One important aspect of the platform is that it allows observability of changes in data and infrastructure. Another is the ability to respond to those changes. The event-driven nature allows us to compose arbitrary preprocessing tasks by having them react to different events.

For inference, we support two ways of serving prediction requests. When a high volume of predictions is required and latency is not an issue, the inference is performed as a batch operation. The model receives large amounts of data, and the predictions are

stored in a database. When low latency predictions are required, machine learning models are deployed as REST endpoints. This type of deployment allows the wearable device to send requests to an endpoint that is always up and receive the prediction immediately. Our implementation fits the latter scenario, and the actual serving is done through the Prediction API service. A runtime container loads the model, and, with little configuration, a web server is set up. The container exposes an endpoint that, when invoked, forwards the request with the necessary parameters to the prediction routine. Once the routine completes, the results are returned as an HTTP response.

Security is a critical aspect of modern cloud-based solutions. A central API gateway sits in front of each publicly accessible service. Public requests can only execute requests to the gateway using an encrypted SSL/TLS channel, and a time-limited authentication token must be provided by clients. This pattern is applied to all services in the solution. All patient-related data stored at rest is automatically encrypted. Encryption is also applied to the homomorphically encrypted data.

4. Results

The main objective of the evaluation is to assess the practical feasibility of the proposed pipeline and to determine the performance impact of using privacy-preserving analysis of wearable sensor data compared to plaintext analysis.

All experiments and evaluations are conducted using the Physionet 2017 Challenge dataset. The proposed classifiers are evaluated with 5-fold cross-validation. In each fold, 80% of the recordings are used to train the privacy-preserving deep neural network on the homomorphically encrypted input data, while the remaining 20% are used for inference. The partitioning of the data splitting is done in such a way that the class distribution between the folds is similar.

The overall performance in terms of accuracy, precision, sensitivity, specificity, and F1-score is computed on the held-out subsets of the data and is listed in Table 1. For a more detailed analysis, the classification performance for each class is shown in Table 2. To tackle the class imbalance problem, we experimented with loss weighting based on class frequency. This strategy is usually used during training to prevent the classifier from being biased towards the most frequent classes. However, the results obtained were comparable to the models trained without this strategy.

Table 1. Overall performance of the proposed classifiers.

Classifier	Performance				
	Accuracy	Precision	Sensitivity	Specificity	F1-Score
MLP	0.83	0.77	0.75	0.92	0.76
1D-CNN	0.79	0.72	0.68	0.90	0.70

Table 2. Performance of the proposed classifiers for each class.

Classifier	Performance							
	Normal		AF		Other		Noise	
	MLP	1D-CNN	MLP	1D-CNN	MLP	1D-CNN	MLP	1D-CNN
Precision	0.88	0.86	0.81	0.70	0.77	0.67	0.63	0.65
Sensitivity	0.92	0.89	0.80	0.66	0.70	0.64	0.59	0.52
Specificity	0.80	0.78	0.98	0.97	0.91	0.87	0.98	0.99
F1-score	0.90	0.87	0.80	0.68	0.73	0.65	0.61	0.58

To compare the performance of the proposed privacy-preserving classifiers with the performance reported in [29], we also analyze the F1 measure. The assessment metric used to rank the models is defined as the mean F1 measure of the classes “Normal”, “AF” and “Other”. A comparison between our privacy-preserving models and the plaintext

approach [29] is summarized in Table 3. For the detection of normal recordings, the average F1 measure is 0.9, for the detection of other rhythms, it is 0.73, while the F1 measure for the classification of atrial fibrillation is close to 0.8. Consequently, the final F1 measure is 0.81.

Table 3. Performance of the proposed methodology using 5-fold cross validation. Comparison with state-of-the-art.

Classifier	Performance							
	F1 Normal		F1 AF		F1 Other		F1 Mean	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
MLP	0.90	0.005	0.80	0.017	0.73	0.012	0.81	0.011
1D-CNN	0.87	0.010	0.68	0.032	0.65	0.026	0.73	0.023
[29]	0.90	0.002	0.79	0.006	0.77	0.004	0.82	0.002

Table 4 lists the runtime performance results obtained with the proposed framework when performing inference on encrypted data. It shows the runtime required to complete the privacy-preserving machine learning-based analysis on a server running Intel(R) Core i7-6700K CPU running at 4.0 GHz (Intel Corporation, Santa Clara, CA, US).

Table 4. Inference time of the proposed classifiers on MORE homomorphically encrypted data.

Classifier	Inference Time on the Server [Seconds]
MLP	Single instance: 0.076 ± 0.024 All data (8528 samples): 7.223 ± 1.248
1D-CNN	Single instance: 1.353 ± 0.268 All data (8528 samples): 2293.831 ± 31.722

5. Discussion

We obtain encouraging results on encrypted data, with the best model achieving a mean F1-score of 0.81, which is comparable to one of the models that won the Physionet 2017 Challenge with a mean F1-score of 0.82 (see Table 3). As shown in Table 1, both privacy-preserving neural network-based models provide satisfactory results, indicating that the proposed approach is capable of performing machine learning-based analysis on personal ECG data without jeopardizing the confidentiality of personal data.

The best results are obtained with the MLP-based classifier trained on encrypted hand-crafter features. Compared to the machine learning-based approach considered in [29], the proposed privacy-preserving MLP-based classifier yields comparable F1-scores for the detection of normal and atrial fibrilation related recordings, but a slightly lower F1-score for the detection of other rhythms (Table 3). As shown in [15], the performance of neural network-based models trained and evaluated on MORE homomorphically encrypted data are statistically indistinguishable from the performance of the corresponding regular model on plaintext data (when the comparison is made under the same conditions). Since the preprocessing step used in the MLP-based approach, including feature extraction, is performed under the same conditions as in [29], the difference in performance is entirely due to the architecture of the machine learning-based classifier. The input features were carefully selected to distinguish atrial fibrillation symptoms and to differentiate the regularity of a clean ECG signal from a random noisy signal. In the work described in [29], the authors leverage this information in their two-stage classification pipeline. However, due to the privacy component, a single multi-class model is proposed herein. Therefore, no distinction can be made between features, nor can prior knowledge be leveraged.

It also appears that using convolutional networks to automatically extract features from the raw ECG signal did not help to improve performance, as shown by the decline in the cross-validation accuracy displayed in Table 1. Although the proposed 1D-CNN deep learning-based model performs worse than the MLP-based approach on all classes, it has the advantage of automatically performing feature extraction during classification. A possible explanation for the drop in performance, compared to the MLP-based classifier,

is the size of the training dataset, as it is commonly shown that deep convolutional neural networks require large data sets to train properly. Since the 1D-CNN model is a fully data-driven model, the classification performance is affected by the quality and quantity of the training data. Therefore, the detection capabilities can be further improved by increasing the number of ECG training samples. Moreover, the performance of the proposed models is also affected by the capabilities of the CipherML library. More specifically, the inability to include max pooling operations, better optimization methods, activation functions that could improve the flow of gradients through the model, the inability to check and compare performance between epochs or stop training when the best model is found are all aspects that limit the performance.

Both models show good performance for the majority of classes but fail to achieve similar performance for the underrepresented classes (Table 2). This is a common behavior observed for models trained on imbalanced datasets. Recent work on the impact of data augmentation on atrial fibrillation classification in short single-lead ECGs [33] shows that although oversampling of minority classes does not significantly improve performance, deep learning models can benefit from synthetically generated data. Although not considered in these experiments, more advanced augmentation techniques could increase the robustness of classification when dealing with imbalanced data. These include synthetic augmentation of the dataset using Synthetic Minority Oversampling Technique (SMOTE) [5] or synthetic ECG signal generating using Generative Adversarial Networks (GANs) [34].

We also observe that all models yield a much lower F1-score when used to classify other rhythm recordings. Due to the lack of actual disease information associated with each recording in this class, identifying the appropriate features becomes more difficult.

We obtained small standard deviations of the F1 scores across testing folds for both privacy-preserving neural network-based models, as presented in Table 3, suggesting that the models produce stable results and that training is not affected by the differences that come from different splits of the training samples.

The advantage of wearable medical devices is that they allow real-time health monitoring. Ultimately, it is not only the performance of the analysis that matters but also the ability to perform it in real time. Although the proposed encryption mechanism is less secure than classical homomorphic encryption, it has the potential to provide results in a runtime suitable for real-time health monitoring applications (Table 4). More specifically, the privacy-preserving AI-based analysis on a single-channel ECG recording lasting for 60 s is performed in less than 2 s.

Although the results are promising, the analysis has some limitations. First, we only experimented with 1D-CNN for the deep learning-based approach, while sequence data-oriented architectures such as RNNs and Temporal Convolutional Networks (TCNs) have been used in the past [35,36]. This limitation is imposed by the current capabilities of the CipherML library. Second, the detection performance needs to be further improved, especially in terms of robustness. Appropriate ways to handle an imbalanced dataset need to be explored. Finally, MORE is a symmetric key-based homomorphic encryption scheme. Although both training and prediction can be performed on MORE encrypted data, training a machine learning-based model on a collection of encrypted samples located on the cloud implies that each sample is encrypted with the same key.

6. Conclusions

We focused on designing a cloud-based platform for developing machine learning-based analytics solutions for wearable sensor data by protecting the integrity of the collected health data. A symmetric-key homomorphic encryption scheme was used for privacy-preserving computation in the cloud within machine learning-based models. We presented CipherML, a C++ framework for training and deploying learning approaches on homomorphically encrypted data.

The proposed privacy-preserving framework is applied to distinguish atrial fibrillation from normal rhythms, other abnormal rhythms, and noise from short single-lead ECG

signals acquired by a wearable device. Two different approaches have been investigated for atrial fibrillation detection: a machine learning-based method using a feature engineering step, and a fully data-driven approach directly using the encrypted raw ECG signal.

In the first scenario, the wearable device collects the ECG signal, extracts the features from the recording, performs encryption, and sends the encrypted feature data to the cloud for machine learning-based classification. However, this is not feasible for devices with limited computation capabilities. Moreover, transmitting only the extracted features has little use outside of a specific predetermined scenario, a considerable limitation that prevents the data being reused for further analytics and the development of other prediction tools.

Therefore, in the second approach, the wearable device encrypts the raw ECG recording and sends it to the cloud for deep-learning-based classification. Although the CNN model achieves a lower mean F1-score than the model trained on hand-crafted features, classification performance can be further improved by increasing the amount of training data. This approach has the primary advantage of removing the requirement to extract problem-specific hand-crafted features, allowing the model to choose what to focus on based on the problem being addressed.

Although the presented algorithms are trained on encrypted data, they achieve a performance comparable to that of existing plaintext solutions. In both scenarios, the input data from the wearable device is encrypted and sent to the cloud for further processing by the AI models. Within seconds, the outcome of the computation is obtained and returned to the user in encrypted form, which can be interpreted only after decryption.

This proof-of-concept study aims to assess the ability of cloud AI-based analytics software solutions to process personal ECG data without compromising privacy. Our AF detection experiments prove that the combination of homomorphic encryption and learning-based models is effective in building privacy-preserving classification systems while maintaining a runtime that is feasible for real-world applications.

Overall, we identify two main limitations of the current study that should be addressed in future research activities. First, the CipherML library has a series of limitations: max-pooling cannot be formulated because no comparison operations can be performed on ciphertexts, the only optimizer currently available is SGD, and advanced training strategies such as gradient clipping, learning rate decay, or early stopping are not supported because they rely on comparison operations.

Second, the MORE encryption scheme is a fundamentally different approach compared to standard FHE schemes based on modulo operations over integers, where security is based on the difficulty of factorizing large numbers. The adopted scheme based on linear operations over continuous variables is known to be vulnerable in cases where there are known pairs of plaintext-ciphertext values with the same key (i.e., chosen ciphertext attack). However, this is not a disabling feature for this particular scenario since the plaintext data are never disclosed or distributed.

On a different note, although privacy and security are critical aspects and should be enforced, compromising data availability and usability will not be a sustainable solution. Data has become an extremely valuable resource, and it is imperative to maximize its collection and distribution. The future of AI-based solutions is beyond doubt dependent on the availability of data, which is more and more becoming a challenge. Personal devices, including wearables, will likely become an increasing supply of data to fuel the immense demand. We strongly believe that homomorphic encryption is going to be the essential tool for achieving such goals. Although it is currently only a supplementary feature, homomorphic encryption is likely to be one of the most important future research directions for AI in general. Besides improving the performance, security, and usability of homomorphic encryption for AI, researchers should focus on a standardized homomorphic encryption methodologies that can become generally accepted by the scientific community, allowing AI and data analysis methods to be built on a strong security framework.

Author Contributions: Conceptualization, A.V., L.M.I., R.M.T, T.S. and C.-I.N.; methodology, A.V., L.M.I., C.-I.N. and R.M.T.; software, A.V., C.-I.N. and R.M.T.; validation, A.V., L.M.I., R.M.T. and C.-I.N.; formal analysis, A.V., C.-I.N., R.M.T., T.S. and L.M.I.; investigation, A.V., C.-I.N., R.M.T., T.S. and L.M.I.; resources, L.M.I. and C.S.; data curation, A.V.; writing—original draft preparation, A.V., L.M.I. and R.M.T.; writing—review and editing, A.V., L.M.I., C.-I.N., R.M.T., T.S. and C.S.; supervision, L.M.I. and R.M.T.; project administration, L.M.I.; funding acquisition, L.M.I.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant of the Romanian Ministry of Education and Research, CCCDI - UEFISCDI, project number PN-III-P2-2.1-PED-2019-2415, within PNCDI III. This work was supported by a FLAG-ERA JTC 2016 grant (CONVERGENCE). This work was supported by a grant of the Ministry of Education and Research, CNCS/CCCDI—UEFISCDI, project number PN-III-P3-3.6-H2020-2020-0145/2021. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 875351.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. Data can be accessed at: https://physionet.org/content/challenge-2017/1.0.0 (accessed on 5 January 2020).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ECG	Electrocardiogram
RF	Random Forest
SVM	Support Vector Machine
MLP	Multi-Layer Perceptron
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
FHE	Fully Homomorphic Encryption
MORE	Matrix Operation for Randomization or Encryption
AF	Atrial Fibrillation
GAN	Generative Adversarial Network
SMOTE	Synthetic Minority Oversampling Technique
TCN	Temporal Convolutional Network
SSL	Secure Sockets Layer
TLS	Transport Layer Security
HTTP	Hypertext Transfer Protocol
API	Application Programming Interface

References

1. El-Gayar, O.; Ambati, L.S.; Nawar, N. Wearables, Artificial Intelligence, and the Future of Healthcare. In *AI and Big Data's Potential for Disruptive Innovation*; Strydom, M., Buckley, S., Eds.; IGI Global: Hershey, PA, USA, 2020; pp. 104–129.
2. Morillo, C.; Banerjee, A.; Perel, P.; Wood, D.; Jouven, X. Atrial fibrillation: The current epidemic. *J. Geriatr. Cardiol. JGC* **2017**, *14*, 195–203. [PubMed]
3. Hall, A.; Mitchell, A.; Wood, L.; Holland, C. Effectiveness of a single lead AliveCor electrocardiogram application for the screening of atrial fibrillation. *Medicine* **2020**, *99*, e21388. [CrossRef] [PubMed]
4. Kaissis, G.; Makowski, M.; Rueckert, D.; Braren, R. Secure, privacy-preserving and federated machine learning in medical imaging. *Nat. Mach. Intell.* **2020**, *2*, 305–311. [CrossRef]
5. Nurmaini, S.; Tondas, A.E.; Darmawahyuni, A.; Rachmatullah, M.N.; Partan, R.U.; Firdaus, F.; Tutuko, B.; Pratiwi, F.; Juliano, A.H.; Khoirani, R. Robust detection of atrial fibrillation from short-term electrocardiogram using convolutional neural networks. *Future Gener. Comput. Syst.* **2020**, *113*, 304–317. [CrossRef]
6. Ebrahimi, Z.; Loni, M.; Daneshtalab, M.; Gharehbaghi, A. A review on deep learning methods for ECG arrhythmia classification. *Expert Syst. Appl. X* **2020**, *7*, 100033.

7. Clifford, G.; Liu, C.; Moody, B.; wei Lehman, L.; Silva, I.; Li, Q.; Johnson, A.E.W.; Mark, R. AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017. In Proceedings of the 2017 Computing in Cardiology (CinC), Rennes, France, 24–27 September 2017; pp. 1–4.
8. Acar, A.; Aksu, H.; Uluagac, S.; Conti, M. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Comput. Surv.* **2017**, *51*, 1–35. [CrossRef]
9. Yousuf, H.; Lahzi, M.; Salloum, S.A.; Shaalan, K. Systematic Review on Fully Homomorphic Encryption Scheme and Its Application. In *Recent Advances in Intelligent Systems and Smart Applications. Studies in Systems, Decision and Control*; Al-Emran M., Shaalan K., Hassanien A., Eds.; Springer: Cham, Switzerland, 2021; pp. 537–551.
10. Orlandi, C.; Piva, A.; Barni, M. Oblivious Neural Network Computing via Homomorphic Encryption. *EURASIP J. Inf. Secur.* **2007**, *2007*, 37343. [CrossRef]
11. Gilad-Bachrach, R.; Dowlin, N.; Laine, K.; Lauter, K.E.; Naehrig, M.; Wernsing, J.R. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016.
12. Chabanne, H.; de Wargny, A.; Milgram, J.; Morel, C.; Prouff, E. Privacy-Preserving Classification on Deep Neural Network. *IACR Cryptol. ePrint Arch.* **2017**, *2017*, 35.
13. Hesamifard, E.; Takabi, H.; Ghasemi, M. CryptoDL: Deep Neural Networks over Encrypted Data. *arXiv* **2017**, arXiv:1711.05189.
14. Juvekar, C.; Vaikuntanathan, V.; Chandrakasan, A. GAZELLE: A Low Latency Framework for Secure Neural Network Inference. In Proceedings of the USENIX Security Symposium, Baltimore, MD, USA, 15–17 August 2018 .
15. Vizitiu, A.; Nita, C.I.; Puiu, A.; Suciu, C.; Itu, L. Applying Deep Neural Networks over Homomorphic Encrypted Medical Data. *Comput. Math. Methods Med.* **2020**, *2020*, 3910250. [CrossRef]
16. Pulido-Gaytan, L.B.; Tchernykh, A.; Cortés-Mendoza, J.M.; Babenko, M.; Radchenko, G.; Avetisyan, A.; Drozdov, A. Privacy-preserving neural networks with Homomorphic encryption: Challenges and opportunities. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 1666–1691. [CrossRef]
17. Microsoft Research. Microsoft SEAL (Release 3.2); Microsoft Research, Redmond, WA, 2019. Available online: https://github.com/Microsoft/SEAL (accessed on 11 May 2020).
18. Fan, J.; Vercauteren, F. Somewhat Practical Fully Homomorphic Encryption. *IACR Cryptol. ePrint Arch.* **2012**, *2012*, 144.
19. Cheon, J.H.; Kim, A.; Kim, M.; Song, Y.S. Homomorphic Encryption for Arithmetic of Approximate Numbers. In Proceedings of the ASIACRYPT 2017, Hong Kong, China, 3–7 December 2017.
20. Halevi, S.; Shoup, V. Algorithms in HElib. *IACR Cryptol. ePrint Arch.* **2014**, *2014*, 106.
21. Brakerski, Z.; Gentry, C.; Vaikuntanathan, V. (Leveled) Fully Homomorphic Encryption without Bootstrapping. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Cambridge, MA, USA, 8–10 January 2012.
22. Lee, J.W.; Kang, H.; Lee, Y.; Choi, W.; Eom, J.; Deryabin, M.; Lee, E.; Lee, J.; Yoo, D.; Kim, Y.S.; No, J.S. Privacy-Preserving Machine Learning with Fully Homomorphic Encryption for Deep Neural Network. *IACR Cryptol. ePrint Arch.* **2021**, *2021*, 783.
23. Folkerts, L.; Gouert, C.; Tsoutsos, N.G. REDsec: Running Encrypted DNNs in Seconds. *IACR Cryptol. ePrint Arch.* **2021**, *2021*, 1100.
24. Ducas, L.; Micciancio, D. FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. In Proceedings of the EUROCRYPT 2015, Sofia, Bulgaria, 26–30 April 2015.
25. Kipnis, A.; Hibshoosh, E. Efficient Methods for Practical Fully Homomorphic Symmetric-key Encrypton, Randomization and Verification. *IACR Cryptol. ePrint Arch.* **2012**, *2012*, 637.
26. Chollet, F. Keras. 2015. Available online: https://github.com/fchollet/keras (accessed on 6 January 2020).
27. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *CoRR* **2015**, arXiv:1412.6980.
28. Goldberger, A.; Amaral, L.A.; Glass, L.; Hausdorff, J.M.; Ivanov, P.; Mark, R.; Mietus, J.; Moody, G.; Peng, C.; Stanley, E. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation* **2000**, *101*, E215–E220. [CrossRef]
29. Datta, S.; Puri, C.; Mukherjee, A.; Banerjee, R.; Choudhury, A.; Singh, R.; Ukil, A.; Bandyopadhyay, S.; Pal, A.; Khandelwal, S. Identifying normal, AF and other abnormal ECG rhythms using a cascaded binary classifier. In Proceedings of the 2017 Computing in Cardiology (CinC), Rennes, France, 24–27 September 2017; pp. 1–4.
30. Teijeiro, T.; Félix, P.; Presedo, J.; Castro, D. Heartbeat Classification Using Abstract Features From the Abductive Interpretation of the ECG. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 409–420. [CrossRef]
31. Ministry for Primary Industries. Accelerate Your Transformation with Google Cloud. 2021. Available online: https://cloud.google.com/ (accessed on 20 June 2021)
32. Bisong, E. Google Cloud Machine Learning Engine (Cloud MLE). In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*; Apress: Berkeley, CA, USA, 2019.
33. Hatamian, F.N.; Ravikumar, N.; Vesal, S.; Kemeth, F.P.; Struck, M.; Maier, A. The Effect of Data Augmentation on Classification of Atrial Fibrillation in Short Single-Lead ECG Signals Using Deep Neural Networks. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 1264–1268.
34. Delaney, A.; Brophy, E.; Ward, T. Synthesis of Realistic ECG using Generative Adversarial Networks. *arXiv* **2019**, arXiv:1909.09150.

35. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Deep learning for time series classification: A review. *Data Min. Knowl. Discov.* **2019**, *33*, 917–963. [CrossRef]
36. Bai, S.; Kolter, Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271.