Privacy Risks in Time Series Forecasting: User- and Record-Level Membership Inference

Nicolas Johansson[‡*], Tobias Olsson[‡*], Daniel Nilsson[†], Johan Östman[†], Fazeleh Hoseini[†]
[*]Chalmers University of Technology [†]AI Sweden

arXiv:2509.04169v2 [cs.LG] 12 Feb 2026

***Abstract*—Membership inference attacks (MIAs) aim to determine whether specific data were used to train a model. While extensively studied on classification models, their impact on time series forecasting remains largely unexplored. We address this gap by introducing two new attacks: (i) an adaptation of multivariate LiRA, a state-of-the-art MIA originally developed for classification models, to the time series forecasting setting, and (ii) a novel end-to-end learning approach called Deep Time Series (DTS) attack. We evaluate our attacks in realistic settings on the TUH-EEG and ELD datasets, targeting two strong forecasting architectures, LSTM and the state-of-the-art N-HiTS, under both record- and user-level threat models. Our results show that forecasting models are vulnerable, with user-level attacks often achieving perfect detection. The proposed methods achieve the strongest performance in several settings, establishing new baselines for privacy risk assessment in time series forecasting. Furthermore, the vulnerability increases with longer prediction horizons and smaller training populations, echoing trends observed in large language models.**

***Index Terms*—Membership Inference Attacks, Time Series, Privacy Auditing, Forecasting Models, Temporal Data**

I. Introduction

The integration of machine learning (ML) into real-world applications has significantly expanded the ways in which industries use data for decision-making. Time series forecasting, in particular, is widely applied in domains such as healthcare, finance, and smart infrastructure. However, when deployed in settings that involve sensitive information, these models raise important concerns regarding data privacy.

Modern ML models have been shown to memorize information from their training data [1]. Recent work demonstrates that memorized information can, in some cases, be extracted by adversaries interacting with the model [2], [3]. These findings have driven interest in developing empirical methods to assess how much sensitive information a model may inadvertently reveal, commonly referred to as privacy auditing [4].

Membership inference attacks (MIAs) constitute the most fundamental building block of privacy auditing. The goal of a MIA is to infer whether a specific data point was part of a model's training set. Although simply inferring membership may not necessary breach privacy, a MIA can be used as a component in more sophisticated attacks, e.g., data reconstruction [5], [6]. Moreover, a successful MIA implies a degree of memorization or overfitting [7] that may expose vulnerabilities towards other types of attacks [8]. Furthermore, MIAs have proven useful in evaluating machine unlearning methods [9], establishing lower bounds on the privacy guarantees of differentially-private algorithms [10], and in identifying dataset contamination [11].

Despite the extensive study of MIAs across domains such as image classification [3], [12], [13], text generation [14]–[16], and graph neural networks [17], [18], their implications for time series forecasting models remain largely unexplored. A few studies have investigated membership inference in the context of aggregated time series data, where an adversary attempts to determine whether an individual's data was included in a published group-level summary [19], [20]. In contrast, our work focuses on predictive models trained with time series data, where the question is whether a given data point was used to train the model.

In time series datasets, data is typically collected continuously from individual users or devices over a period of time. To make such data suitable for different machine learning tasks, it is commonly segmented into fixed-length sequences. This process results in multiple overlapping or adjacent segments, i.e., training records, derived from the same user. Consequently, each user contributes multiple data points to the training set. This is similar to the case of some text analysis tasks, where a single document is partitioned into smaller units, such as paragraphs or sentences [21], [22]. In this work, we distinguish MIAs targeting single-record membership from those targeting user-level (multi-record) membership.

At a deeper level, it remains unclear what the appropriate unit of analysis should be in MIAs. Is it sufficient to assess privacy risk at the level of individual records, or should the focus shift to user-level exposure, given that multiple records may collectively reveal the presence of a single user in the training data? Similar concerns have been raised in the context of large language models (LLMs) where the quantity of interest may vary from sequence level to entire datasets [23].

MIAs against time series forecasting were recently studied at the record level [24], demonstrating that unique characteristics of time series, e.g., seasonality and trend, can be exploited by MIAs. While this study provides valuable insights, it falls short of providing a comprehensive analysis of how various forecasting models respond to MIAs, and omits critical details on the performance of these models, e.g., the degree of overfitting, and the data processing. Moreover, against classification models, their ensemble method does not perform

[‡]These authors contributed equally.

This work was supported by Vinnova, the Swedish innovation agency, under grants 2023-03000 and 2025-03066.

on par with recent state-of-the-art MIAs such as the Likelihood Ratio Attack (LiRA) [12] or the Robust Membership Inference Attack (RMIA) [3]. These attacks may also be effective for time series forecasting: LiRA's recent extension to LLMs demonstrates how correlations in sequential loss trajectories can be exploited [25].

This paper makes the following contributions:[1]

- We devise two novel attacks: a statistics-based one that builds on multivariate LiRA, and a novel classifier-based one, coined Deep Time Series (DTS), which achieves state-of-the-art performance in the record-level setting.
- We provide the first systematic investigation of state-of-the-art MIAs in the context of time series forecasting. In particular, we adapt two state-of-the-art membership inference attacks, LiRA and RMIA, originally developed for classification tasks, to the forecasting setting by incorporating time series-specific signals.
- We conduct systematic experiments across realistic forecasting architectures, real-world time series datasets, and two adversarial threat models, targeting both record-level and user-level MIAs.

II. Related Work

A. Membership Inference Attacks

Membership inference attacks were first introduced in early work [1] and have since been extensively studied [2], [3], [12], [13], [18], [26]. They are commonly divided into statistics-based and classifier-based approaches. Statistics-based methods [2], [3], [12], [18] compute metrics from the outputs of the target and auxiliary shadow models, which are trained to replicate the target's behavior. Building on this idea, state-of-the-art variants refine the attack by calibrating thresholds to the difficulty of each sample using shadow models [3], [12], [18]. A key limitation, however, is that they often fail to distinguish members from non-members when their outputs are too similar [13].

In contrast, classifier-based attacks [1] train a separate classifier to distinguish between the target model's behavior on training and non-training inputs. To improve discrimination, some approaches train this classifier on features obtained by mimicking the target model's training process at different stages [13], [26], allowing the attack to differentiate between data points with similar target model outputs.

B. Quantity of Interest in MIAs

Most of the MIA literature focuses on individual records. However, in some contexts, such as copyright disputes or dataset contamination, the goal may instead be to infer the membership of a collection of records [11], [27]. In the context of text, for instance, quantities of interest may include sequences, paragraphs, documents, or even entire datasets [23]. Earlier works often apply record-level MIAs and then aggregate the results across multiple records to produce a membership score for the collection. This approach has proven more effective than inferring membership for individual records [21], [22].

Most existing collection-level studies assume that the adversary has direct access to some of the training records, a setting referred to as user-level membership inference. In contrast, Kandpal et al. [28] consider a user inference setting, where the attacker does not have access to exact training samples but only to other data points drawn from the same distribution. In our work, we focus on the former setting, user-level membership inference.

C. MIAs on Time Series Forecasting Models

MIAs have primarily been studied in the context of image classification and text generation, while their application to time series models remains underexplored. However, recent work has leveraged time series-specific properties, such as loss trajectories, to design classifier-based attacks [13], [26]. For instance, Li et al. [26] extract features such as fluctuation, correlation, and decline rate from the loss trajectory, which are then used to train a classifier that distinguishes between member and non-member samples. Similarly, the loss trajectory of text sequences was used together with LiRA to attack LLMs, using more than 400 shadow models [25].

Recent work has examined attacks against time series forecasting models [24]. Their main contribution is the introduction of two time series-specific attack features: trend and seasonality. Seasonality is captured using a 2D Discrete Fourier Transform (2D-DFT), which extracts frequency components across time and variables. The features include Fourier coefficients of the predicted and true sequences, along with their L2 difference. For trend, a low-degree polynomial is fitted to each sequence, and the resulting features are the polynomial coefficients and their L2 difference.

III. Preliminaries

A. Signals as Attack Features

In the context of membership inference against time series models, various signals can be extracted from model outputs to distinguish between training and non-training examples. We outline the core signals explored in prior work and our study.

1) Error-Based Signals: Standard forecasting metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and Symmetric Mean Absolute Percentage Error (SMAPE) quantify discrepancies between the predicted and true future sequences [29]. While MSE and MAE are scale-dependent, SMAPE is scale-invariant and bounded between 0 and 1.

2) Structural Signals: Trend and seasonality are commonly used to characterize high-level temporal structure in time series data. Koren et al. [24] leverage these components to capture broader temporal patterns for membership inference attacks, as discussed in Section II-C.

3) Representation-Based Signals: TS2Vec [30] offers a contrastive learning framework that produces rich, multiscale embeddings of time series. Using a shared encoder and hierarchical contrastive loss, TS2Vec learns representations that can

[1]The code is available at https://github.com/aidotse/LeakPro/tree/SaTML-time-series-mia.

generalize across tasks. In our attack context, the L2 distance between the TS2Vec embeddings of predicted and ground-truth sequences acts as a discriminative feature.

B. Time Series Forecasting

We consider a dataset composed of time series samples collected from U distinct entities, each potentially of different length. Let T_u denote the time series associated with entity $u \in [U]$. From T_u, we construct n_u training records

$$\mathcal{D}_u = \{(X_{u,1}, Y_{u,1}), \ldots, (X_{u,n_u}, Y_{u,n_u})\}.$$

Each record $j \in [n_u]$ consists of an input sequence

$$X_{u,j} = [x_{t_j - L+1}, \ldots, x_{t_j}] \in \mathbb{R}^{M \times L},$$

where L is the look-back window length, M is the number of variables, and t_j denotes the index of the last observation in the input sequence, i.e., the time step at the training–label split, and an output sequence (target label)

$$Y_{u,j} = [y_{t_j+1}, \ldots, y_{t_j+H}] \in \mathbb{R}^{M \times H},$$

where H is the prediction horizon. The complete dataset is given as

$$\mathcal{D} = \bigcup_{u \in [U]} \mathcal{D}_u.$$

The goal is to train a model f_θ, parameterized by θ, to predict H future time steps given a sequence $X \in \mathbb{R}^{M \times L}$, i.e.,

$$\hat{Y} = f_\theta(X) \in \mathbb{R}^{M \times H}.$$

We refer to the (randomized) training process as $\mathcal{T}$, such that $\theta = \mathcal{T}(\mathcal{D}_{\text{train}})$, where $\mathcal{D}_{\text{train}} \subseteq \mathcal{D}$.

IV. Threat Model and Setup

We adopt a game-based formulation of MIAs [2], [8], modeling the attack as an adversarial game between a challenger and an attacker. We assume that each entity $u \in [U]$ samples their data from a distribution π_u. Hence, the user dataset $\mathcal{D}_u$ contains records $X_{u,j} \sim \pi_u$ for $j \in [n_u]$. We denote a dataset $\mathcal{D}_u$, of size n_u, sampled i.i.d. from π_u as $\mathcal{D}_u \sim \pi_u^{n_u}$. Moreover, we denote by π the distribution over records across all users, referred to as the record population, to which the adversary has sampling access.

To instantiate the MIA game, the challenger samples a set of I users $\mathcal{I}_{\text{train}} \subset [U]$ uniformly at random. For each $u \in \mathcal{I}_{\text{train}}$, the challenger samples a dataset $\mathcal{D}_u \sim \pi_u^{n_u}$ and concatenates these to form the training set

$$\mathcal{D}_{\text{train}} = \bigcup_{u \in \mathcal{I}_{\text{train}}} \mathcal{D}_u.$$

Here, we set $n_u = n$ for all u. Analogously, a validation dataset $\mathcal{D}_{\text{val}}$ is constructed from users in $\mathcal{I}_{\text{val}} \subset [U] \setminus \mathcal{I}_{\text{train}}$. The remaining users are collected in $\mathcal{I}_{\text{test}} = [U] \setminus (\mathcal{I}_{\text{train}} \cup \mathcal{I}_{\text{val}})$, and a corresponding test dataset $\mathcal{D}_{\text{test}}$ is created in the same way. Finally, the challenger trains a model f_θ, where $\theta = \mathcal{T}(\mathcal{D}_{\text{train}})$.

The adversary has black-box access to the model f_θ, i.e., it can obtain the forecast for arbitrary input queries but does not have access to the weights θ. Furthermore, the adversary is assumed to know the training procedure, including the hyperparameters and model architecture of the target model.

Game 1 Record-Level Membership Inference Game

Input: Population distribution π, number of users in the population U, user population distributions $\{\pi_u\}_{u \in [U]}$, training algorithm $\mathcal{T}$, number of users for training I, dataset size n.
Output: Adversary's guess $\tilde{b}$.

1: Sample I users into $\mathcal{I}_{\text{train}}$ uniformly at random from $[U]$
2: Create training dataset $\mathcal{D}_{\text{train}} \leftarrow \cup_{u \in \mathcal{I}_{\text{train}}} \mathcal{D}_u$, $\mathcal{D}_u \sim \pi_u^n$
3: Train model parameters: $\theta \leftarrow \mathcal{T}(\mathcal{D}_{\text{train}})$
4: Sample member record $(X_1, Y_1) \sim \mathcal{D}_{\text{train}}$
5: Sample non-member record $(X_0, Y_0) \sim \mathcal{D}_{\text{test}}$
6: Sample random bit $b \sim \text{Bernoulli}(0.5)$
7: Challenger sends $(\mathcal{T}, f_\theta, (X_b, Y_b))$ to the adversary
8: Adversary infers membership $\tilde{b} \leftarrow \mathcal{A}(\mathcal{T}, \pi, f_\theta, (X_b, Y_b))$

Game 2 User-Level Membership Inference Game

Input: Population distribution π, number of users in the population U, user population distributions $\{\pi_u\}_{u \in [U]}$, training algorithm $\mathcal{T}$, number of users for training I, dataset size n, challenge dataset size n_{c}.
Output: Adversary's guess $\tilde{b}$

1: Sample I users into $\mathcal{I}_{\text{train}}$ uniformly at random from $[U]$
2: Create training dataset $\mathcal{D}_{\text{train}} \leftarrow \cup_{u \in \mathcal{I}_{\text{train}}} \mathcal{D}_u$, $\mathcal{D}_u \sim \pi_u^n$
3: Train model parameters: $\theta \leftarrow \mathcal{T}(\mathcal{D}_{\text{train}})$
4: Sample a member user $u_1 \sim \mathcal{I}_{\text{train}}$
5: Sample a non-member user $u_0 \sim \mathcal{I}_{\text{test}}$
6: Sample a random bit $b \sim \text{Bernoulli}(0.5)$
7: Sample a challenge dataset $\tilde{\mathcal{D}}_{u_b} \sim \pi_{u_b}^{n_c}$ of size $n_{\text{c}} \leq n$
8: Challenger sends $(\mathcal{T}, f_\theta, \tilde{\mathcal{D}}_{u_b})$ to the adversary
9: Adversary infers membership $\tilde{b} \leftarrow \mathcal{A}(\mathcal{T}, \pi, f_\theta, \tilde{\mathcal{D}}_{u_b})$

We distinguish between record-level and user-level MIA. In record-level MIA, the adversary goal is to devise an attack $\mathcal{A}$ to infer if a given data record (X, Y) was included in the training set $\mathcal{D}_{\text{train}}$ of f_θ. The privacy game instantiation is illustrated in Game 1. Notably, the adversary does not have access to the weights θ but can only query the model f_θ, and may use the record-population π in the attack.

In user-level MIAs, formalized in Game 2, the adversary attempts to determine whether a collection of records belonging to a single user was included in the training set, thereby inferring the membership status of that user. We note that, to setup the attack, the adversary has access to the record population, π, rather than the user populations $\{\pi_u\}_{u \in [U]}$, i.e., it is unable to sample from a given user distributions.

V. Method

We propose two attacks for time series forecasting models: a statistics-based and a classifier-based. The attacks are described from the perspective of record-level attacks, and their extension to user-level attacks is discussed in Section V-C. Both attacks rely on shadow models that are trained on datasets

created in the same way as the target dataset but using different user-splits. We denote the ith shadow model by ϕ_i.

A. Multi-Signal LiRA

As discussed in Section III-A, a variety of statistical signals can be extracted from time series forecasts to serve as features for membership inference. Building on this, we propose a multi-signal variant of LiRA, inspired by prior work that uses likelihood scores from multiple augmentations of the same record to improve inference performance [12, Sec. IV.C].

Let $\mathcal{S}$ denote the set of available attack signals. For a given record (X, Y), the model predicts $\hat{Y} = f_\theta(X)$, from which a signal is computed according to $s(Y, \hat{Y})$, $s \in \mathcal{S}$. Note that some signals in $\mathcal{S}$ may not make use of $\hat{Y}$, in which case it is simply ignored.

For the attack, we train K shadow models to mimic the target model f_θ. The adversary runs each record through both the target model f_θ and each shadow model f_{ϕ_i}, $i \in [K]$, to yield $K+1$ predictions. Let $s(Y, f_{\phi_i}(X))$ denote the value of signal s evaluated on the output of shadow model i. We compute the mean and standard deviation for each signal type over the shadow models. Shadow models are divided into in-models, which include (X, Y) in their training data, and out-models, which do not. For each group, we compute the mean and standard deviation of the signal values, yielding (μ_v^s, σ_v^s) where $v \in \{\text{in}, \text{out}\}$ and $s \in \mathcal{S}$ [12].

We model the signals for each group v as multivariate Gaussian distributed, $\mathbf{s} \sim \mathcal{N}(\mu_v, \Sigma_v)$, where $\mu_v = (\mu_v^s)_{s\in\mathcal{S}}$ and Σ_v is the empirical covariance matrix of the signals in group v. A principled approach would be to estimate the full covariance matrix Σ_v, thereby capturing dependencies between signals. However, stable estimation of the off-diagonal entries of Σ_v typically requires a large number of shadow models; prior work on text models has used more than 400 to achieve reliable estimates [25]. Given the practical constraint that we can only train a limited number of shadow models, we approximate Σ_v by a diagonal matrix containing only the variances $(\sigma_v^s)^2$ for each $s \in \mathcal{S}$.

For the online attack, where shadow models may include the target data, the resulting likelihood-ratio score is

$$\text{score}(X, Y; f_\theta) = \frac{p\big(\mathbf{s}(X, Y; f_\theta) \,\big|\, \mathcal{N}(\mu_{\text{in}}, \Sigma_{\text{in}})\big)}{p\big(\mathbf{s}(X, Y; f_\theta) \,\big|\, \mathcal{N}(\mu_{\text{out}}, \Sigma_{\text{out}})\big)}, \tag{1}$$

where $\mathbf{s}(X, Y; f_\theta) = \big(s(Y, f_\theta(X))\big)_{s\in\mathcal{S}}$ is the vector of signal values for record (X, Y).

For the offline attack, the score is defined as the multivariate Gaussian tail probability under the out-model distribution:

$$\text{score}(X, Y; f_\theta) = \Pr\big(\mathbf{Z} \preceq \mathbf{s}(X, Y; f_\theta)\big), \mathbf{Z} \sim \mathcal{N}\big(\mu_{\text{out}}, \Sigma_{\text{out}}\big), \tag{2}$$

where $\preceq$ denotes element-wise inequality. When Σ_{out} is diagonal, this reduces to the product of per-signal univariate tail probabilities.

B. Deep Time Series Attack

MIAs for time series forecasting typically rely on manually engineered statistical features extracted from model outputs, including our proposed attack in Section V-A. These approaches require domain-specific feature selection and may fail to capture subtle patterns present in the raw temporal structure of predictions and their corresponding errors. To address this limitation, we propose the DTS Attack, a classifier-based approach that bypasses manual signal extraction. DTS directly learns to infer membership from model outputs by training a deep-learning classifier. The procedure is formalized in Algorithm 1.

The attack is based on supervised training of a classifier that learns to distinguish members from non-members given the output of a forecasting model. To construct the training dataset, we assume the adversary has access to an auxiliary dataset $\mathcal{D}_{\text{aux}}$. In the online setting, $\mathcal{D}_{\text{aux}}$ constitutes all of $\mathcal{D}$ whereas, in the offline setting, $\mathcal{D}_{\text{aux}}$ contains no records from $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}} \cup \mathcal{D}_{\text{test}}$. From $\mathcal{D}_{\text{aux}}$, K (potentially overlapping) datasets $\mathcal{D}_i$ are sampled and used to train shadow models $\{\phi_i\}_{i=1}^K$ (Step 1 in Algorithm 1).

To construct the training dataset $\mathcal{D}_{\text{DTS}}$, for each shadow model ϕ_i, $i \in [K]$, we randomly sample a fraction f of records from $\mathcal{D}_{\text{aux}}$. Each sampled record (X, Y) is evaluated by ϕ_i to produce the triplet $(Y, f_{\phi_i}(X), \ell_i)$, where $\ell_i = \mathbf{1}\{(X, Y) \in \mathcal{D}_i\} \in \{0, 1\}$ indicates whether (X, Y) was used to train ϕ_i. These triplets are added into $\mathcal{D}_{\text{DTS}}$, with each entry in $(\mathbb{R}^{M\times H} \times \mathbb{R}^{M\times H} \times \{0, 1\})$, yielding a dataset of size $Kf|\mathcal{D}_{\text{aux}}|$.

The membership classifier θ_{DTS} is then trained on $\mathcal{D}_{\text{DTS}}$ to map the $(Y, \hat{Y})$ pairs into a membership probability,

$$\theta_{\text{DTS}} : \big(\mathbb{R}^{M\times H} \times \mathbb{R}^{M\times H}\big) \longrightarrow [0, 1],$$

as shown in Steps 2–3 of Algorithm 1. Through this training process, the classifier learns to distinguish between training and non-training records by leveraging temporal dependencies and statistical regularities in the joint distribution of model predictions and true outcomes.

After training, θ_{DTS} can be used to attack the target model f_θ. Given a target record (X, Y), the target model prediction $\hat{Y} = f_\theta(X)$ is computed, after which $(Y, \hat{Y})$ is passed to θ_{DTS} to obtain the membership inference score

$$\text{score}\big(X, Y; f_\theta\big) = \theta_{\text{DTS}}\big(Y, \hat{Y}\big), \tag{3}$$

which represents the estimated probability that (X, Y) was part of the training set of the target model. The attacker then compares this probability to a decision threshold to determine membership.

C. Record-Level to User-Level MIA

The two attacks described in the previous sections output *record-level* membership inference scores, i.e., the estimated probability that a given record from the audit dataset was included in the target model's training set. Extending this formulation to the user level introduces an additional challenge, as each user typically contributes multiple records. User-level membership decisions must therefore take into account several record-level scores from the same user, which may be combined in various ways.

Algorithm 1 Deep Time Series (DTS) MIA

Input: Target model f_θ, auxiliary dataset $\mathcal{D}_{\text{aux}}$, sampling fraction f, training algorithm $\mathcal{T}$, number of shadow models K, target data (X, Y)
Output: Membership score for the target data (X, Y)
Step 1: Train shadow models
for $i = 1 \dots K$ **do**
  Sample a training subset $\mathcal{D}_i$ from $\mathcal{D}_{\text{aux}}$
  Train shadow model $\phi_i \leftarrow \mathcal{T}(\mathcal{D}_i)$
end for
Step 2: Construct DTS training dataset $\mathcal{D}_{\text{DTS}}$
$\mathcal{D}_{\text{DTS}} = \emptyset$
for $i = 1 \dots K$ **do**
  Sample a fraction f from $\mathcal{D}_{\text{aux}}$ into $\tilde{\mathcal{D}}$
  for each $(\tilde{X}, \tilde{Y}) \in \tilde{\mathcal{D}}$ **do**
    Compute prediction $\hat{Y} = f_{\phi_i}(\tilde{X})$
    Assign $\ell_i = 1$ if $\tilde{X} \in \mathcal{D}_i$, otherwise $\ell_i = 0$
    Add tuple $(\tilde{Y}, \hat{Y}, \ell_i)$ to $\mathcal{D}_{\text{DTS}}$
  end for
end for
Step 3: Train the classifier
Train θ_{DTS} using $\mathcal{D}_{\text{DTS}}$
Step 4: Score the target record
Compute target prediction $\hat{Y} = f_\theta(X)$
Obtain score: $\text{score}\big(X, Y; f_\theta\big) = \theta_{\text{DTS}}\big(Y, \hat{Y}\big)$

Several combination strategies can be used, including simple majority voting, averaging or taking the median of record-level scores, or emphasizing extreme outliers. In this work, we adopt a likelihood-based approach, computing the product of the record-level membership probabilities for a given user. Under the simplifying assumption that record-level predictions are independent, this product corresponds to the overall likelihood that the user's dataset was included in training.

Formally, a user $u \in \mathcal{I}_{\text{train}}$ contributes $\mathcal{D}_u = \{(X_{u,1}, Y_{u,1}), \dots, (X_{u,n}, Y_{u,n})\}$, consisting of n training records. During an attack on user u, the adversary observes a collection of n_c records $\tilde{\mathcal{D}}_u \sim \pi_u^{n_c}$ whose elements may coincide with those in $\mathcal{D}$. The user-level membership score for user u is then given as

$$\text{score}_{\text{user}}\Big(\tilde{\mathcal{D}}_u; f_\theta\Big) = \prod_{(X,Y)\in\tilde{\mathcal{D}}_u} \text{score}\left(X, Y; f_\theta\right), \qquad (4)$$

where $\text{score}(X, Y; f_\theta)$ is the record-level membership probability for a single forecasting record (X, Y).

VI. Experimental Setup

A. Data Setup and Preprocessing

We evaluate our attacks on two real-world time series datasets from privacy-sensitive domains, each requiring domain-specific preprocessing. The first consists of electroencephalogram (EEG) signals which are biometric by nature and highly distinctive across individuals, making them inherently identifiable [31]. The second, electricity load data, while non-biometric, can still reveal extensive private information: prior work has shown that fine-grained consumption traces can be exploited to infer household activities, device usage, occupancy patterns, and even socio-economic or employment status [32]. We describe both datasets in detail below.

1) The Temple University Hospital EEG Data Corpus (TUH-EEG v2.0.1): The TUH-EEG data corpus is a clinical dataset containing 26846 EEG recordings from around 1500 individuals, collected between 2002 and 2017 [33]. The majority of the EEG recordings contains 31 channels, however, some have as few as 20. Most signals were sampled at 250 Hz (87%), with smaller portions recorded at 256 Hz (8.3%), 400 Hz (3.7%), and 512 Hz (1%).

To ensure data quality, we selected a subset of 100 recordings from the first 300 individuals with recordings sampled at 250 Hz. We removed the initial 60 seconds of each recording which often contain artifacts such as calibration waves, and excluded sessions with persistent noise or flat channels.

Following prior work [24], we selected $M = 3$ frontal EEG leads (FP1, FP2, F3) and truncated recordings to 15000 time steps. Records were generated using a sliding window with a lookback of $L = 100$, using a stride of 1, and a prediction horizon $H = 20$, see Section III-B.

All time series were normalized using interquartile range (IQR) scaling via the `RobustScaler` from scikit-learn [34], which we found to perform best empirically due to its robustness against outliers and inter-subject variability.

2) Electricity Load Diagrams (ELD) Dataset: The ELD dataset [35] is a widely used benchmark for time series forecasting. It is based on recordings of electricity consumption from 370 Portuguese households over a four-year period (2011–2014), with a sampling resolution of 15 minutes. Measurements are expressed in kilowatts. Since not all households were monitored throughout the entire period, some recordings contain zero-padding to make all time series equal length.

To prepare the data, we removed the zero-padding and aggregated the time resolution to 1 hour by summing every four consecutive measurements. Households with fewer than 15000 valid time steps or with average usage below 200 kW/h or above 2000 kW/h were excluded to avoid unreliable signals and extreme outliers. We then randomly selected 100 households for our experiments. Each series was truncated to 15000 time steps and normalized using IQR scaling, following the same procedure used for EEG data.

The dataset constitutes a univariate time series, i.e., $M = 1$, and records were generated via a sliding window, using a stride of 1, with a lookback of $L = 100$ and a prediction horizon of $H = 20$, as described in Section III-B.

B. Data Split

After preprocessing, each dataset contains time series data from $I = 100$ individuals. For each individual, the window operation results in 14881 data points per individual. We adopt an individual-based partitioning strategy to ensure that no individual's data appears in both the train and test datasets. In particular, we divide the 100 individuals into a training

dataset $\mathcal{D}_{\text{train}}$, validation dataset $\mathcal{D}_{\text{val}}$, and test dataset $\mathcal{D}_{\text{test}}$ of 20 individuals each. We emphasize that we make use of the validation dataset for early-stopping to limit the amount of overfitting, hence, making the attack setting more difficult. The remaining 40 individuals constitute the auxiliary dataset $\mathcal{D}_{\text{aux}}$ and is used by the adversary to train shadow models. During the attack, we make use of $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$ to sample records to infer the membership of. In the online setting, each shadow model is trained on a randomly selected 50% subset of individuals from $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$ instead of sampling from $\mathcal{D}_{\text{aux}}$ as in the offline setting. For user-level attacks, we let $\tilde{\mathcal{D}}_u = \mathcal{D}_u$, i.e., the adversary observes all user data records during the attack. We consider partial observations in the ablations.

C. Target Model Architectures

We assess membership inference risks in time series forecasting using two representative deep learning architectures: Long Short-Term Memory (LSTM) [36] and Neural Hierarchical Interpolation for Time Series (N-HiTS) [37]. N-HiTS is a hierarchical, multi-resolution forecasting model that extends N-BEATS [38] with multi-rate input sampling and multi-scale interpolation, achieving higher accuracy and efficiency than prior methods in long-horizon forecasting while supporting multivariate and exogenous inputs.

By selecting these two models, we cover both an established baseline and a state-of-the-art approach to time series modeling. Both are implemented in PyTorch and configured for multivariate sequences with varying input lengths and prediction horizons. Their specific configurations and hyperparameters are detailed below.

1) LSTM: The LSTM model is a standard multivariate sequence-to-sequence model. It consists of two unidirectional LSTM layers, each with 64 hidden units. The model accepts sequences of length L with M variables as input (i.e., $X \in \mathbb{R}^{M \times L}$), where the number of input features is set to M. The final hidden state from the second LSTM layer is passed to a fully connected layer with $H \times M$ output neurons, representing the forecast for the next H time steps.

2) N-HiTS: For the N-HiTS model, we build on the official implementation [39], applying minimal modifications to fit our experimental setup. Specifically, we remove dependencies on exogenous variables and adjust input dimension handling for compatibility with our preprocessed datasets. The architecture consists of three stacks, each containing a single block with two fully connected layers of 512 units. Each block employs the `IdentityBasis` function to directly map outputs into backcast and forecast components, without explicitly modeling trends or seasonality. We adopt the default hyperparameters recommended by the original authors: Xavier normal initialization [40], no weight sharing between blocks, no dropout or batch normalization, and ReLU activations. Linear interpolation is used as the pooling method across resolution levels.

D. Training Protocol

We trained the target models using the Adam optimizer with a default learning rate of 0.001 and the MAE loss as the optimization criterion. Training was performed for a maximum of 50 epochs with early stopping based on validation loss, using a patience of 3 epochs. A batch size of 1024 was used for all training runs.

E. Target Model Evaluation

To evaluate the performance of our target models, we report four forecasting metrics: MSE, MAE, SMAPE, and normalized deviation (ND) [38]. MSE and MAE are standard pointwise metrics, while SMAPE and ND measure errors relative to the magnitude of the true values. Given our individual-level data partitioning and the substantial variance across subjects, these scale-sensitive metrics provide more informative indicators of overfitting and generalization. Formally, for a prediction horizon H and M target variables, SMAPE and ND are defined as:

$$\text{SMAPE}(Y, \hat{Y}) = \frac{1}{HM} \sum_{i=1}^{H} \sum_{j=1}^{M} \frac{|Y_{i,j} - \hat{Y}_{i,j}|}{|Y_{i,j}| + |\hat{Y}_{i,j}|} \tag{5}$$

and

$$\text{ND}(Y, \hat{Y}) = \frac{\sum_{i=1}^{H} \sum_{j=1}^{M} |Y_{i,j} - \hat{Y}_{i,j}|}{\sum_{i=1}^{H} \sum_{j=1}^{M} |Y_{i,j}|}. \tag{6}$$

F. Attacks Implementation

We evaluate attack performance using three state-of-the-art baselines, and our proposed attacks from Section V, employing 64 shadow models for all configurations. Their specific instantiations in our experimental setup are described below.

All attacks leverage the signals introduced in Section III-A. For some attacks, the underlying signal must be unbounded; to accommodate this, we define a rescaled version of SMAPE:

$$\text{rSMAPE} = \log\left(\frac{\text{SMAPE}(Y, \hat{Y})}{1 - \text{SMAPE}(Y, \hat{Y})}\right). \tag{7}$$

1) Baseline 1 (LiRA): We conduct multiple instances of the LiRA attack [12], substituting the rescaled logit from the original formulation with each of the signals introduced in Section III-A and in (7). These variants are evaluated across all datasets and target models in both online and offline settings. Variance for the in- and out-distributions is estimated following the per-example variance method [12, Sec. VI.B].

2) Baseline 2 (RMIA): RMIA [3] compares the likelihood ratio between a model trained with the target record and one where it is replaced by a random population record, using its tail probability as the test score. The attack instantiations use six signals: MSE, MAE, SMAPE, Trend, Seasonality, and TS2Vec, evaluated in both online and offline settings across all datasets and target models. Since rSMAPE is unbounded, it is omitted from RMIA. For RMIA, which operates on probabilities, we apply a monotonically decreasing mapping

$$g(s(X, Y; f_\theta)) = \frac{1}{1 + s(X, Y; f_\theta)}$$

to convert signal values into probability scores. RMIA also requires sampling from the population to estimate the average

shadow-model output; for this, we draw records from the dataset excluding all train, test, and validation instances. We set $\gamma = 1.0$ and $\alpha = 1/3$ for the offline setting [3, App. B.2.2].

3) Baseline 3 (Ensemble Attack): Shachor et al. [41] introduce an Ensemble attack, which assumes the adversary has access to a dataset of in-members (target training data points) along with non-members. The method trains multiple small, specialized classifiers on disjoint subsets of data, each balanced with 50% members and non-members, to improve membership inference. For each subset, several classifiers are trained on target model outputs, or derived features, evaluated, and the best one retained; repeating this across many subsets yields an ensemble whose score for a point is the fraction of models predicting it as a member. This approach is adopted by Koren et al. [24], who are the first to study time series forecasting in this context. In our work, we use the same attack parameters as [24], apply mean aggregation within the ensemble as in [41], and select the best model based on ROC-AUC performance. Specifically, we set the number of complete executions of the attack on different random records to five, the number of train/test repetitions per subset pair to three, the subset size to 50, and the number of fixed member–non-member subset combinations per instance to nine.

Next, we consider the attacks proposed in Section V.

4) Multi-Signal LiRA: We applied the Multi-Signal LiRA attack (Section V-A) to all datasets and target models under both online and offline evaluation settings. The signal vector comprises MSE, MAE, rSMAPE, Trend, Seasonality, and TS2Vec. We assumed a diagonal covariance matrix and estimate each entry using the per-example variance method of [12, Sec. VI.B].[2]

5) Deep Time Series Attack: The DTS attack is applied to all datasets and target models under both online and offline evaluation settings. For each configuration, we construct $\mathcal{D}_{\text{DTS}}$ by sampling $f = 10\%$ from the data set governing the creation of shadow datasets. We select a small f to demonstrate that the method performs strongly even at low values; the performance is expected to improve as f increases. Each DTS classifier is trained with the Adam optimizer, using early stopping (patience $= 3$) and a maximum of 64 epochs.

We evaluate two classifier architectures: a simple LSTM-based model and InceptionTime [42], a convolution-based architecture that, along with its variants, is considered state-of-the-art for time series classification on the UCR archive [43] which is a collection of multiple time series datasets [44]. In contrast to prior work [42], which relies on an ensemble of predictors, we simplify the approach by employing a single one.

[2]We also experimented with using a full covariance matrix, but the limited number of shadow models in our setting prevented stable estimation. While [25] apply a similar idea to LLMs, stabilizing covariance estimation via shrinkage with over 400 shadow models, we intentionally restrict our setup to a scale that remains feasible across the multiple datasets, models, and attack variants considered in this study.

VII. Experimental Results

A. Target Models

Table I reports target model performance on the TUH-EEG and ELD datasets across training, validation, and test splits, with results averaged over five runs to account for variability from random initialization and dataset partitioning. The randomization includes both the target model's data split and the user selection; in each run, a new subset of users is sampled (100 out of 300 for TUH-EEG and 100 out of 370 for ELD). The similar performance gaps between splits for both architectures indicate that early stopping effectively prevents overfitting. Across all metrics (MSE, MAE, SMAPE, and ND), N-HiTS outperforms LSTM. On TUH-EEG, the largest gains for N-HiTS are observed in MSE (a reduction from 4.15×10^{-9} to 8.43×10^{-10}) and ND (from 0.319 to 0.223) on the test set. On ELD, improvements are most pronounced in MSE (from 1.85×10^{4} to 1.37×10^{4}) and SMAPE (from 0.0754 to 0.0645). MSE and MAE show higher variance across runs, while the scale-invariant SMAPE and ND remain more stable, making them better suited for cross-dataset comparison.

B. Attack Results

The performance of the different attacks is presented in Table II for the TUH-EEG dataset and Table III for the ELD dataset, and in Table IV. We follow earlier works [3], [12], [18] and report the true positive rate (TPR) at a given FPR (Table II and Table III) and the ROC-AUC score in Table IV. For the complete ROC curves, we refer to Appendix A. In particular, for record-level attacks, we report TPR at an FPR of 0.1% and 0.01% while, for user-level attacks, we report the TPR at 0%. All user-level results correspond to an access ratio of $g = 1$. All results are averaged over five independent runs, with both the mean and standard deviation reported. For each attack type, we illustrate the strongest configuration with boldface and the overall best attack for each reported metric is highlighted in and for the online and offline setting, respectively.

1) Data Sensitivity: Attacks are consistently more effective on TUH-EEG than on ELD, a difference likely driven by several factors. First, our EEG experiments use three input channels versus a single channel for ELD, so with the lookback and horizon fixed, more input information is available to the MIA. Second, as discussed in Section VI-A, EEG signals are highly individual-specific, whereas electricity usage, though also carrying identifiable patterns, is less distinctive due to overlapping daily routines and shared household behaviors. Finally, EEG data has much finer temporal resolution (4ms) compared to ELD (1h), providing attackers with far more granular input. Together, these factors help explain the stronger attack performance observed on EEG-trained models, as forecasting models naturally adapt to stable user-specific temporal structure when it is present.

2) Comparing Attack Strategies: The Ensemble attack [41] performs poorly under the stringent metrics considered, ranking near the bottom across datasets and settings. Its limited

Datasets	Models	Metrics	Train	Validation	Test
TUH-EEG	LSTM	MSE	6.68×10^{-10} $\pm 4.46 \times 10^{-10}$	4.42×10^{-9} $\pm 4.87 \times 10^{-9}$	4.15×10^{-9} $\pm 2.20 \times 10^{-9}$
		MAE	1.28×10^{-5} $\pm 2.77 \times 10^{-6}$	2.05×10^{-5} $\pm 5.29 \times 10^{-6}$	2.30×10^{-5} $\pm 4.45 \times 10^{-6}$
		SMAPE	0.247 $\pm$0.0285	0.271 $\pm$0.0186	0.285 $\pm$0.0111
		ND	0.198 $\pm$0.0323	0.296 $\pm$0.0590	0.319 $\pm$0.0063
	N-HiTS	MSE	2.60×10^{-10} $\pm 1.47 \times 10^{-10}$	6.97×10^{-10} $\pm 3.04 \times 10^{-10}$	8.43×10^{-10} $\pm 5.10 \times 10^{-10}$
		MAE	8.92×10^{-6} $\pm 1.43 \times 10^{-6}$	1.49×10^{-5} $\pm 2.45 \times 10^{-6}$	1.60×10^{-5} $\pm 2.46 \times 10^{-6}$
		SMAPE	0.208 $\pm$0.0301	0.258 $\pm$0.0195	0.267 $\pm$0.0048
		ND	0.139 $\pm$0.0196	0.216 $\pm$0.0294	0.223 $\pm$0.0201
ELD	LSTM	MSE	1.47×10^{4} $\pm 6.00 \times 10^{3}$	2.77×10^{4} $\pm 1.90 \times 10^{4}$	1.85×10^{4} $\pm 6.65 \times 10^{3}$
		MAE	65.8 $\pm$12.9	88.0 $\pm$25.0	73.8 $\pm$6.9
		SMAPE	0.0607 $\pm$0.0178	0.0668 $\pm$0.0226	0.0754 $\pm$0.0056
		ND	0.0984 $\pm$0.0177	0.116 $\pm$0.0315	0.123 $\pm$0.0089
	N-HiTS	MSE	7.92×10^{3} $\pm 2.80 \times 10^{3}$	1.95×10^{4} $\pm 1.29 \times 10^{4}$	1.37×10^{4} $\pm 3.45 \times 10^{3}$
		MAE	49.1 $\pm$9.6	69.5 $\pm$19.6	61.1 $\pm$10.9
		SMAPE	0.0471 $\pm$0.0164	0.0556 $\pm$0.0250	0.0645 $\pm$0.0088
		ND	0.0731 $\pm$0.0126	0.0926 $\pm$0.0306	0.102 $\pm$0.0143

TABLE I: Performance of the target models on the TUH-EEG and ELD datasets. Each entry reports the mean ± standard deviation across five runs with different random seeds for Train, Validation, and Test splits (unscaled metrics).

per-model training data and lack of adaptability likely explain its weak performance.

RMIA [3] stands out in several scenarios. It achieves the highest user-level performance on TUH-EEG across both models and is the strongest method in the online setting for N-HiTS on ELD. At the record level, RMIA dominates offline attacks for N-HiTS on TUH-EEG and for all models on ELD. However, in the online record-level setting, RMIA is frequently outperformed by LiRA, Multi-Signal LiRA, and DTS. Considering ROC-AUC in Table IV, RMIA emerges as the clear winner in the offline setting.

For both datasets, LiRA performs competitively at the user level, sometimes matching the best results, but generally lags behind the strongest methods at the record-level, especially in the offline setting. In the online setting, TS2Vec often yields LiRA's best record-level results, while rSMAPE tends to perform better offline. No single signal dominates across all cases, and Trend and SMAPE are typically the weakest. A notable example is TUH-EEG with N-HiTS in the online setting, where all signals except Trend achieve a perfect 100% user-level TPR at 0% FPR across all seeds. On ELD, LiRA's record-level scores are often surpassed by DTS or RMIA, but it remains competitive at the user level.

Our proposed Multi-Signal LiRA improves over the best single-signal LiRA in many online record-level settings but rarely outperforms the top-performing non-LiRA methods. However, with respect to ROC-AUC, Table IV, it achieves the best performance against an LSTM target model. In the offline setting, it can underperform compared to the strongest individual signals, such as rSMAPE and TS2Vec. We anticipate that a fully multivariate LiRA, able to model correlations between signals, could further improve performance, as the signals are anticipated to correlate. However, because our current implementation models the covariance matrix as diagonal, due to the limited number of shadow models, these correlations cannot be fully exploited.

The DTS attack offers a strong deep learning–based alternative, achieving some of the highest online record-level TPRs across both datasets, particularly with the InceptionTime architecture. For example, on TUH-EEG with N-HiTS, DTS reaches 23.32% TPR at 0.1% FPR in the online setting, outperforming all other methods by a substantial margin. With respect to ROC-AUC, it achieves the strongest performance against the N-HiTS architecture, achieving up to 0.884. DTS is less competitive in the offline setting, where RMIA often dominates, but still achieves respectable results. Its performance depends heavily on the membership inference classifier architecture, with InceptionTime often outperforming LSTM.

In Appendix B, we present results on the attack performance when the forecast models are trained using differential privacy.

3) Offline vs Online Attacks: Across all evaluations, online attacks consistently outperformed their offline counterparts, with especially large gains at the user level. This gap is substantially larger than what has been reported for other data modalities [3], [12], [18], suggesting that membership inference in time series forecasting may be particularly sensitive to distributional shifts. A likely factor is our individual-based partitioning. With only 100 diverse users, temporal and user-specific patterns can introduce significant distribution differences between the train/test and auxiliary sets, in contrast to the record-level splits commonly used in prior work. We further explore this hypothesis in Section VII-C.

For both datasets, the strongest record-level results in the online setting are obtained by LiRA and DTS, while RMIA and LiRA dominate at the user level. The magnitude of the online–offline gap can be striking. For example, on TUH-EEG with N-HiTS at FPR $= 0.01\%$, LiRA achieves a TPR of 2.64% online compared to only 0.13% offline. The difference is even more pronounced for DTS, which reaches 15.81% online compared to 0.67% offline.

4) User vs Record-Level Attacks: Our results show a consistent gap between record-level and user-level MIAs; it is substantially easier to infer membership status of a user than an individual record. This is in line with previous works [21], [22] that argue for user-level attacks, that aggregate predictions across multiple records, providing stronger signals for

		N-HiTS			LSTM		
		Record-level		User-level	Record-level		User-level
Attacks	Signal/Model	0.1% FPR	0.01% FPR	0% FPR	0.1% FPR	0.01% FPR	0% FPR
Ensemble	MSE	0.00 ±0.00	0.00 ±0.00	6.00 ±7.35	0.00 ±0.00	0.00 ±0.00	2.00 ±2.45
	MAE	0.00 ±0.00	0.00 ±0.00	8.00 ±8.12	0.00 ±0.00	0.00 ±0.00	7.00 ±5.10
	SMAPE	0.00 ±0.00	0.00 ±0.00	6.00 ±2.00	0.00 ±0.00	0.00 ±0.00	5.00 ±5.48
	rSMAPE	0.00 ±0.00	0.00 ±0.00	8.00 ±4.00	0.00 ±0.00	0.00 ±0.00	**10.00** ±7.07
	Trend	**0.03** ±0.05	0.00 ±0.00	4.00 ±5.83	**0.01** ±0.01	0.00 ±0.00	5.00 ±4.47
	Seasonality	**0.03** ±0.03	0.00 ±0.00	9.00 ±8.00	0.00 ±0.00	0.00 ±0.00	4.00 ±4.90
	TS2Vec	0.00 ±0.00	0.00 ±0.00	**15.00** ±14.83	0.00 ±0.00	0.00 ±0.00	1.00 ±2.00
RMIA-Online	MSE	3.28 ±2.02	1.38 ±1.98	**100.0** ±0.00	1.90 ±1.92	0.16 ±0.24	32.00 ±26.94
	MAE	**3.40** ±1.91	**1.48** ±1.93	**100.0** ±0.00	**2.02** ±2.12	0.36 ±0.37	**41.00** ±25.38
	SMAPE	1.70 ±1.57	0.46 ±0.53	**100.0** ±0.00	1.40 ±2.27	1.06 ±2.11	36.00 ±23.54
	Trend	1.01 ±1.43	0.38 ±0.68	58.00 ±21.12	1.19 ±1.90	0.58 ±1.11	10.00 ±7.07
	Seasonality	2.08 ±1.56	0.74 ±1.09	**100.0** ±0.00	1.49 ±1.84	0.56 ±0.90	30.00 ±21.68
	TS2Vec	2.61 ±1.80	1.31 ±1.98	**100.0** ±0.00	1.84 ±2.30	**1.28** ±2.23	37.00 ±23.37
RMIA-Offline	MSE	1.24 ±2.25	0.00 ±0.00	**42.00** ±30.59	1.51 ±1.24	0.66 ±1.32	17.00 ±14.35
	MAE	1.10 ±2.06	**1.04** ±2.09	40.00 ±29.50	1.67 ±1.17	0.60 ±1.21	15.00 ±10.49
	SMAPE	2.20 ±1.58	0.00 ±0.00	27.00 ±18.60	1.24 ±1.39	0.01 ±0.01	12.00 ±6.78
	Trend	0.46 ±0.28	0.03 ±0.05	9.00 ±4.90	0.38 ±0.41	0.02 ±0.03	10.00 ±5.48
	Seasonality	**2.37** ±2.19	0.97 ±1.45	14.00 ±10.68	1.87 ±1.90	**0.73** ±1.39	8.00 ±8.12
	TS2Vec	0.29 ±0.59	0.00 ±0.00	41.00 ±25.38	**1.94** ±1.27	0.00 ±0.00	**19.00** ±12.41
LiRA-Online	MSE	4.02 ±2.81	0.19 ±0.24	**100.0** ±0.00	3.05 ±2.35	2.27 ±2.03	22.00 ±10.77
	MAE	6.73 ±2.60	1.35 ±1.79	**100.0** ±0.00	3.43 ±2.47	2.58 ±2.19	28.00 ±15.03
	SMAPE	5.04 ±2.20	1.11 ±1.10	**100.0** ±0.00	3.19 ±2.37	1.93 ±2.08	34.00 ±12.41
	rSMAPE	6.65 ±2.39	2.26 ±1.83	**100.0** ±0.00	3.50 ±2.40	2.23 ±2.13	**35.00** ±13.42
	Trend	5.19 ±4.12	2.42 ±2.05	94.00 ±4.90	2.26 ±2.51	1.99 ±2.39	15.00 ±5.48
	Seasonality	7.28 ±2.50	1.77 ±2.12	**100.0** ±0.00	3.42 ±2.41	**2.61** ±2.13	23.00 ±12.08
	TS2Vec	**7.41** ±0.97	**2.64** ±1.49	**100.0** ±0.00	**3.57** ±2.19	2.57 ±2.19	31.00 ±9.70
LiRA-Offline	MSE	0.01 ±0.02	0.00 ±0.00	2.00 ±2.45	1.66 ±2.16	1.35 ±2.13	4.00 ±4.90
	MAE	0.07 ±0.10	0.00 ±0.00	6.00 ±8.00	1.97 ±2.35	**1.80** ±2.29	6.00 ±3.74
	SMAPE	0.20 ±0.11	0.05 ±0.03	16.00 ±5.83	1.88 ±2.25	1.54 ±1.90	**8.00** ±6.00
	rSMAPE	**1.04** ±0.93	0.12 ±0.08	**22.00** ±9.80	**2.12** ±2.38	1.69 ±1.98	**8.00** ±4.00
	Trend	0.55 ±0.57	**0.13** ±0.14	12.00 ±8.72	1.84 ±2.14	1.46 ±1.98	3.00 ±4.00
	Seasonality	0.08 ±0.11	0.01 ±0.01	10.00 ±7.07	1.97 ±2.34	1.71 ±2.22	6.00 ±3.74
	TS2Vec	0.33 ±0.34	0.05 ±0.03	8.00 ±6.78	1.90 ±2.24	1.71 ±2.22	5.00 ±3.16
Multi-Signal LiRA-Online		9.11 ±2.86	2.71 ±2.88	100.0 ±0.00	3.79 ±2.51	2.99 ±2.35	32.00 ±8.72
Multi-Signal LiRA-Offline		0.12 ±0.16	0.01 ±0.00	10.00 ±8.94	2.12 ±2.46	1.88 ±2.35	5.00 ±3.16
DTS-Online	LSTM	**23.32** ±6.65	15.62 ±7.22	**96.00** ±4.90	**2.97** ±3.75	**2.15** ±2.69	**30.00** ±23.45
	InceptionTime	22.39 ±7.80	**15.81** ±7.35	90.00 ±6.32	2.49 ±4.99	1.70 ±3.39	20.00 ±19.24
DTS-Offline	LSTM	**2.09** ±3.13	**0.67** ±1.01	**27.00** ±16.61	0.17 ±0.16	0.00 ±0.01	**9.00** ±15.62
	InceptionTime	1.12 ±1.45	0.64 ±1.28	22.00 ±23.79	**0.23** ±0.29	**0.05** ±0.10	5.00 ±7.75

TABLE II: Membership inference attack performance on the TUH-EEG dataset. We report record-level TPR at fixed FPRs (0.1% and 0.01%) and user-level TPR at 0% FPR. Results are averaged over five runs, with mean and standard deviation shown. For attacks with multiple configurations, the best-performing configuration is bolded. The and indicate the best-performing attack over each metric in the online and offline setting, respectively.

membership inference. Crucially, this discrepancy highlights a potential pitfall in evaluation since only considering record-level metrics may significantly underestimate the potential privacy risk. For instance, in the online setting against the N-HiTS model on the TUH-EEG dataset, LiRA achieves just 7.41% TPR at 0.1% FPR at the record-level, but 100% TPR at 0% FPR at the user level, indicating a far more severe vulnerability when considering membership on user level.

Another observation is the large variance observed in some user-level results. As shown in Section VII-C, this variance decreases with an increasing number of individuals, suggesting improved stability in larger populations. It is worth noting that user-level performance is reported at 0% FPR, a statistically sensitive threshold. Because it allows no false positives, even

		N-HiTS			LSTM		
		Record-level		User-level	Record-level		User-level
Attacks	Signal/Model	0.1% FPR	0.01% FPR	0% FPR	0.1% FPR	0.01% FPR	0% FPR
Ensemble Attack	MSE	0.01 ±0.02	0.00 ±0.00	8.00 ±6.78	0.00 ±0.00	0.00 ±0.00	4.00 ±4.90
	MAE	0.00 ±0.00	0.00 ±0.00	6.00 ±7.35	0.00 ±0.00	0.00 ±0.00	5.00 ±5.48
	SMAPE	**0.02** ±0.02	0.00 ±0.00	2.00 ±4.00	**0.02** ±0.05	0.00 ±0.00	3.00 ±6.00
	rSMAPE	0.00 ±0.00	0.00 ±0.00	3.00 ±4.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00
	Trend	0.01 ±0.02	0.00 ±0.00	5.00 ±4.47	0.00 ±0.00	0.00 ±0.00	2.00 ±2.45
	Seasonality	0.00 ±0.00	0.00 ±0.00	**9.00** ±9.70	0.00 ±0.00	0.00 ±0.00	**6.00** ±3.74
	TS2Vec	0.00 ±0.00	0.00 ±0.00	2.00 ±4.00	0.00 ±0.00	0.00 ±0.00	3.00 ±4.00
RMIA-Online	MSE	1.12 ±1.26	**0.31** ±0.37	44.00 ±20.35	**1.16** ±0.58	**0.15** ±0.14	**19.00** ±17.44
	MAE	**1.19** ±1.20	0.26 ±0.40	59.00 ±19.60	1.00 ±0.60	0.13 ±0.15	16.00 ±12.41
	SMAPE	0.67 ±0.44	0.09 ±0.12	47.00 ±21.59	0.24 ±0.21	0.04 ±0.05	16.00 ±11.58
	Trend	0.17 ±0.12	0.02 ±0.04	21.00 ±7.35	0.09 ±0.06	0.01 ±0.01	13.00 ±11.66
	Seasonality	0.97 ±0.69	0.04 ±0.05	**73.00** ±19.65	0.46 ±0.37	0.08 ±0.10	**19.00** ±14.97
	TS2Vec	0.79 ±0.50	0.03 ±0.05	64.00 ±30.23	0.40 ±0.47	0.10 ±0.19	18.00 ±11.66
RMIA-Offline	MSE	0.06 ±0.13	0.00 ±0.00	7.00 ±6.00	1.28 ±1.33	0.26 ±0.23	6.00 ±5.83
	MAE	0.50 ±0.66	0.00 ±0.01	**8.00** ±6.78	**1.29** ±1.44	0.03 ±0.06	5.00 ±6.32
	SMAPE	0.02 ±0.04	0.00 ±0.00	**8.00** ±2.45	0.61 ±0.45	0.00 ±0.00	4.00 ±4.90
	Trend	0.34 ±0.15	0.03 ±0.06	6.00 ±5.83	0.16 ±0.09	0.02 ±0.03	6.00 ±5.83
	Seasonality	**0.89** ±0.86	**0.31** ±0.50	7.00 ±6.00	0.68 ±1.02	**0.42** ±0.84	5.00 ±6.32
	TS2Vec	0.61 ±0.90	0.03 ±0.04	7.00 ±5.10	0.77 ±1.03	0.36 ±0.73	**7.00** ±7.48
LiRA-Online	MSE	1.06 ±0.57	0.28 ±0.21	43.00 ±28.21	2.26 ±3.18	0.84 ±1.19	29.00 ±20.10
	MAE	1.99 ±1.00	0.65 ±0.36	49.00 ±26.72	3.29 ±2.64	1.47 ±0.88	35.00 ±15.49
	SMAPE	1.50 ±0.61	0.46 ±0.23	43.00 ±26.76	2.18 ±1.46	0.78 ±0.29	29.00 ±11.58
	rSMAPE	1.94 ±0.79	0.59 ±0.26	**53.00** ±29.43	3.10 ±1.75	0.93 ±0.51	**38.00** ±12.08
	Trend	1.34 ±0.29	0.45 ±0.20	50.00 ±19.49	1.17 ±0.72	0.31 ±0.32	27.00 ±19.65
	Seasonality	2.02 ±0.85	0.76 ±0.37	49.00 ±26.72	3.46 ±3.07	1.43 ±1.14	34.00 ±13.56
	TS2Vec	**2.16** ±1.08	**0.88** ±0.45	48.00 ±25.42	**3.50** ±2.86	**1.68** ±1.25	33.00 ±12.08
LiRA-Offline	MSE	0.07 ±0.07	0.01 ±0.01	4.00 ±5.83	0.03 ±0.03	0.00 ±0.00	3.00 ±4.00
	MAE	0.26 ±0.17	0.07 ±0.07	9.00 ±7.35	0.21 ±0.24	0.02 ±0.02	7.00 ±4.00
	SMAPE	0.35 ±0.23	0.09 ±0.06	10.00 ±7.07	0.18 ±0.14	0.05 ±0.06	8.00 ±7.48
	rSMAPE	**0.59** ±0.27	0.13 ±0.10	**12.00** ±9.27	0.34 ±0.26	0.07 ±0.04	**9.00** ±4.90
	Trend	0.22 ±0.10	0.05 ±0.03	6.00 ±3.74	0.06 ±0.08	0.01 ±0.02	5.00 ±4.47
	Seasonality	0.30 ±0.18	0.08 ±0.09	10.00 ±8.37	0.24 ±0.18	0.07 ±0.09	8.00 ±5.10
	TS2Vec	0.52 ±0.36	**0.26** ±0.23	9.00 ±5.83	**0.41** ±0.32	**0.10** ±0.06	7.00 ±5.10
Multi-Signal LiRA-Online		2.32 ±1.01	0.85 ±0.41	48.00 ±26.76	4.15 ±4.13	2.29 ±2.27	34.00 ±12.81
Multi-Signal LiRA-Offline		0.32 ±0.20	0.08 ±0.08	10.00 ±6.32	0.37 ±0.42	0.06 ±0.06	8.00 ±4.00
DTS-Online	LSTM	6.34 ±3.91	2.77 ±1.87	56.00 ±22.45	**4.78** ±4.56	**2.57** ±2.42	**34.00** ±21.54
	InceptionTime	**6.66** ±3.48	**2.94** ±1.69	**65.00** ±21.21	4.49 ±4.83	2.40 ±3.20	24.00 ±25.77
DTS-Offline	LSTM	0.57 ±0.39	0.15 ±0.18	**13.00** ±12.88	0.76 ±0.85	0.26 ±0.39	**9.00** ±5.83
	InceptionTime	**0.68** ±0.71	**0.21** ±0.26	11.00 ±13.19	**0.78** ±0.86	**0.30** ±0.47	6.00 ±4.90

TABLE III: Membership inference attack performance on the ELD dataset. We report record-level TPR at fixed FPRs (0.1% and 0.01%) and user-level TPR at 0% FPR. Results are averaged over five runs, with mean and standard deviation shown. For attacks with multiple configurations, the best-performing configuration is bolded. The and indicate the best-performing attack over each metric in the online and offline setting, respectively.

minor fluctuations in model predictions can lead to significant changes in the measured TPR. Despite this instability, we report results at 0% FPR to highlight that some attacks achieve perfect (100%) user-level detection under this strict setting.

C. Ablation Study

We conducted three ablation studies to examine how forecasting horizon, population size, and access ratio influence attack performance. The first two studies used an N-HiTS target model on the ELD dataset, while the third was performed on the EEG dataset, where user-level attacks achieve their strongest performance. Across all studies, we evaluated three representative online attacks: LiRA with TS2Vec, Multi-Signal LiRA, and DTS with InceptionTime.

1) Impact of Forecasting Horizon: Figure 1 reports record-level and user-level TPRs at fixed FPRs for varying forecasting horizons $H \in \{5, 10, 20, 40, 80\}$. Longer horizons generally yield higher record-level TPRs, indicating increased information leakage with larger prediction windows. This trend is consistent across all attacks. At the user level, DTS continues

Attacks	Signal/Model	TUH-EEG		ELD	
		N-HiTS	LSTM	N-HiTS	LSTM
Ensemble Attack	MSE	0.546 ±0.0823	0.530 ±0.0420	0.504 ±0.0908	**0.509** ±0.0764
	MAE	0.532 ±0.0909	**0.532** ±0.0525	0.503 ±0.0709	0.500 ±0.0767
	SMAPE	0.491 ±0.0476	0.495 ±0.0312	0.496 ±0.0377	0.482 ±0.0711
	rSMAPE	0.494 ±0.0454	0.499 ±0.0392	0.493 ±0.0385	0.484 ±0.0648
	Trend	0.537 ±0.0291	0.506 ±0.0300	**0.509** ±0.0622	0.496 ±0.0599
	Seasonality	0.538 ±0.0911	0.529 ±0.0483	0.484 ±0.0905	0.502 ±0.0719
	TS2Vec	**0.555** ±0.1110	0.507 ±0.0795	0.477 ±0.0599	0.499 ±0.0750
RMIA-Online	MSE	0.744 ±0.0201	0.602 ±0.0302	0.646 ±0.0454	0.658 ±0.0362
	MAE	0.775 ±0.0142	0.614 ±0.0384	**0.665** ±0.0470	**0.669** ±0.0322
	SMAPE	0.747 ±0.0089	**0.614** ±0.0233	0.628 ±0.0554	0.636 ±0.0207
	Trend	0.610 ±0.0121	0.527 ±0.0122	0.581 ±0.0190	0.559 ±0.00678
	Seasonality	0.778 ±0.0282	0.594 ±0.0593	0.660 ±0.0483	0.666 ±0.0312
	TS2Vec	**0.782** ±0.0173	0.609 ±0.0515	0.648 ±0.0414	0.654 ±0.0316
RMIA-Offline	MSE	0.724 ±0.0392	0.631 ±0.0395	0.592 ±0.0326	0.593 ±0.0466
	MAE	0.744 ±0.0443	**0.637** ±0.0406	0.593 ±0.0515	0.580 ±0.0498
	SMAPE	0.665 ±0.0334	0.588 ±0.0265	0.600 ±0.0135	0.593 ±0.0358
	Trend	0.588 ±0.0203	0.551 ±0.0267	0.550 ±0.0534	0.535 ±0.0394
	Seasonality	0.706 ±0.0423	0.622 ±0.0262	0.558 ±0.0767	0.537 ±0.0699
	TS2Vec	**0.748** ±0.0488	0.634 ±0.0416	**0.607** ±0.0223	**0.599** ±0.0360
LiRA-Online	MSE	0.811 ±0.0099	0.640 ±0.0333	0.672 ±0.0445	0.672 ±0.0519
	MAE	0.816 ±0.0102	0.646 ±0.0306	0.682 ±0.0433	**0.681** ±0.0444
	SMAPE	0.796 ±0.0089	0.647 ±0.0186	0.660 ±0.0366	0.661 ±0.0366
	rSMAPE	0.801 ±0.0100	0.650 ±0.0202	0.668 ±0.0361	0.666 ±0.0297
	Trend	0.697 ±0.0204	0.552 ±0.0159	0.646 ±0.0169	0.602 ±0.0397
	Seasonality	0.821 ±0.0112	0.646 ±0.0315	**0.682** ±0.0437	0.680 ±0.0437
	TS2Vec	**0.822** ±0.0147	**0.655** ±0.0374	0.669 ±0.0434	0.672 ±0.0495
LiRA-Offline	MSE	0.440 ±0.0284	0.445 ±0.0321	0.510 ±0.0689	0.461 ±0.0840
	MAE	0.495 ±0.0288	0.474 ±0.0355	0.523 ±0.0558	0.469 ±0.0780
	SMAPE	0.521 ±0.0196	0.486 ±0.0257	0.495 ±0.0146	0.461 ±0.0398
	rSMAPE	**0.563** ±0.0213	**0.518** ±0.0304	**0.532** ±0.0149	**0.488** ±0.0414
	Trend	0.470 ±0.0364	0.459 ±0.0548	0.492 ±0.0654	0.477 ±0.0632
	Seasonality	0.495 ±0.0305	0.476 ±0.0381	0.521 ±0.0568	0.469 ±0.0833
	TS2Vec	0.506 ±0.0182	0.480 ±0.0231	0.500 ±0.0331	0.464 ±0.0405
Multi-Signal LiRA-Online		0.847 ±0.0119	0.661 ±0.0312	0.703 ±0.0439	0.700 ±0.0464
Multi-Signal LiRA-Offline		0.492 ±0.0278	0.469 ±0.0378	0.517 ±0.0567	0.459 ±0.0812
DTS-Online	LSTM	**0.884** ±0.0303	**0.637** ±0.0378	0.754 ±0.0131	**0.667** ±0.069
	InceptionTime	0.866 ±0.0389	0.597 ±0.0540	**0.758** ±0.0137	0.658 ±0.081
DTS-Offline	LSTM	0.653 ±0.0288	**0.529** ±0.0355	**0.574** ±0.0395	0.549 ±0.0312
	InceptionTime	**0.660** ±0.0503	0.504 ±0.0507	0.56 ±0.0218	**0.55** ±0.0265

TABLE IV: Membership inference attack performance on the TUH-EEG and ELD dataset. We report the ROC-AUC averaged over five runs, with mean and standard deviation shown. For attacks with multiple configurations, the best-performing configuration is bolded. The and indicate the best-performing attack in the online and offline setting, respectively.

to improve with longer horizons, whereas LiRA-based attacks peak at around horizon 20 before declining, possibly due to signal degradation or shifts in the underlying data distribution. An increase in attack performance with sequence length has previously been documented in the context of LLMs [14], [45].

2) Impact of Number of Individuals: To examine the effect of population size, we varied the total number of users within $I \in \{50, 100, 200\}$, keeping all other parameters fixed. Each configuration was evaluated over five random seeds to account for variation due to data splits. These totals correspond to $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{test}}$ with 20, 40, and 80 individuals, respectively. As shown in Figure 2, both record-level and user-level TPRs decrease as the number of individuals increases, suggesting that larger populations reduce vulnerability to MIAs under individual-based partitioning.

3) Impact of Partially Observed User Data: We study the effect of the adversary's observation size by letting $\tilde{\mathcal{D}}_u$ consist of subsets of $\mathcal{D}_u$ and varying n_c as $n_c \in \{0, 0.05n, 0.10n, \ldots, n\}$. Here, $n_c = 1$ reflects record-level inference with no user-level context, whereas $n_c = n$ corresponds to the full-observation setting analyzed. Both of these settings are studied in Section VII-B.

In Figure 3, we report the user-level performance, averaged over five independent random subsets using the same target and shadow models. As expected, attack performance increases with n_c; more surprisingly, though, strong results

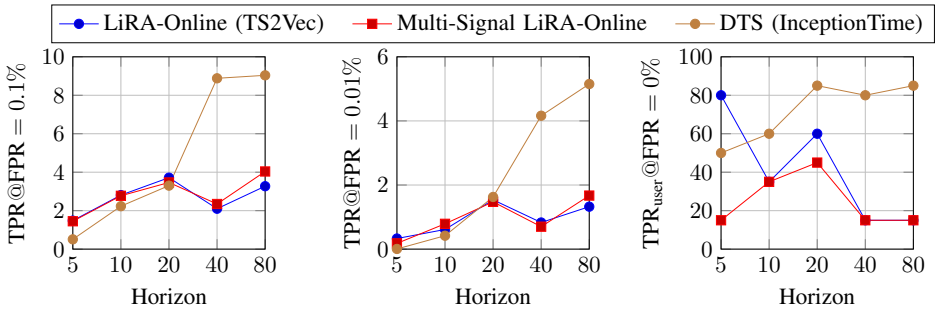


Fig. 1: Effect of forecasting horizon on record-level (left and mid) and user-level (right) TPRs for three attack configurations against the N-HiTS model on the ELD dataset.

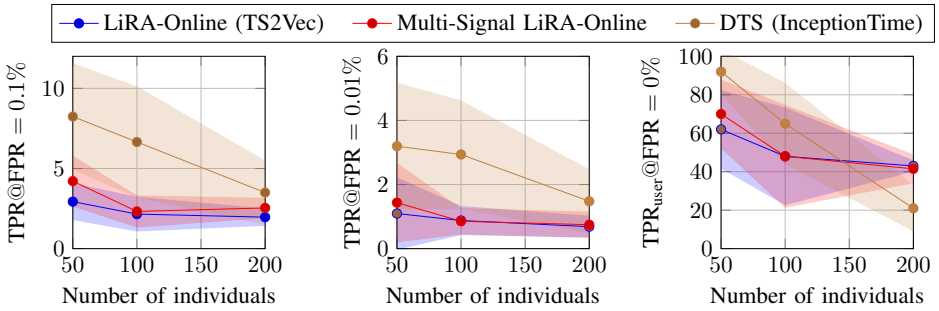


Fig. 2: Effect of varying number of individuals on record-level (left and mid) and user-level TPRs (right) for LiRA-Online, Multi-Signal LiRA-Online, and DTS with InceptionTime; plotted with the confidence interval over five runs.

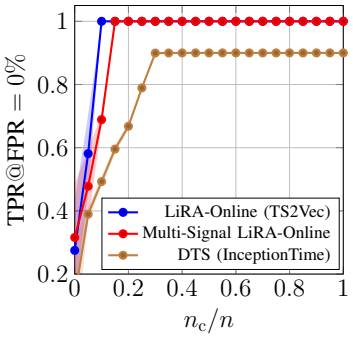


Fig. 3: Effect of observed dataset size on user-level MIA for LiRA-Online, Multi-Signal LiRA-Online, and DTS-Online.

emerge even at small fractions of a user's records, indicating a high susceptibility to user-level membership inference.

VIII. Conclusion

In conclusion, we bring membership inference attacks to deep learning–based time series forecasting, adapting LiRA and RMIA, introducing a multi-signal LiRA, and proposing the deep learning–based DTS attack. Evaluations on real datasets (EEG and ELD) with LSTM and N-HiTS show that these models are vulnerable, particularly under user-level threat models, despite early stopping and good generalization. Online attacks substantially outperform offline ones, and that user-level attacks can reveal severe vulnerabilities missed by record-level evaluation. Ablations further show that longer forecasting horizons increase leakage, while larger populations reduce it. Our results establish strong baselines and highlight key factors influencing privacy risks in time series forecasting, informing the design of both future attacks and defenses.

While our study provides a starting point, it is limited to two datasets and two model architectures. Expanding evaluations to other domains (e.g., ECG signals, financial transactions, IoT sensors, GPS traces), alternative architectures such as transformers and temporal convolutional networks, and settings incorporating exogenous variables or probabilistic forecasts could yield a more complete picture of vulnerabilities. Another important limitation is the relatively small number of individuals in our datasets, which can introduce instability in the results; extending evaluations to larger cohorts would help assess the robustness and generalizability of the findings. Moreover, refining user-level inference, moving beyond independent score aggregation to end-to-end approaches, remains an important open challenge. Addressing these directions is key to understanding privacy risks in time series forecasting.

Ethics Statement

This work aims to improve risk assessment for sensitive time series data by identifying vulnerabilities in forecasting models. This work relies on public datasets and the code is open-sourced to support reproducible research.

REFERENCES

[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE symposium on security and privacy (SP)*, 2017.

[2] J. Ye, A. Maddi, S. K. Murakonda, V. Bindschaedler, and R. Shokri, "Enhanced membership inference attacks against machine learning models," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.

[3] S. Zarifzadeh, P. Liu, and R. Shokri, "Low-cost high-power membership inference attacks," in *International Conference on Machine Learning (ICML)*, 2024.

[4] S. K. Murakonda and R. Shokri, "Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning," in *Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*, 2020.

[5] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models," in *USENIX security symposium (USENIX Security)*, 2021.

[6] J. Zhang, D. Das, G. Kamath, and F. Tramèr, "Position: Membership inference attacks cannot prove that a model was trained on your data," in *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025.

[7] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *USENIX security symposium (USENIX security)*, 2019.

[8] A. Salem, G. Cherubin, D. Evans, B. Köpf, A. Paverd, A. Suri, S. Tople, and S. Zanella-Béguelin, "Sok: Let the privacy games begin! a unified treatment of data inference privacy in machine learning," in *IEEE Symposium on Security and Privacy (SP)*, 2023.

[9] Z. Di, S. Yu, Y. Vorobeychik, and Y. Liu, "Adversarial machine unlearning," in *International Conference on Learning Representations (ICLR)*, 2025.

[10] M. Nasr, S. Songi, A. Thakurta, N. Papernot, and N. Carlin, "Adversary instantiation: Lower bounds for differentially private machine learning," in *IEEE Symposium on security and privacy (SP)*, 2021.

[11] Y. Oren, N. Meister, N. S. Chatterji, F. Ladhak, and T. Hashimoto, "Proving test set contamination in black-box language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[12] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramér, "Membership inference attacks from first principles," in *IEEE Symposium on Security and Privacy (SP)*, 2022.

[13] Y. Liu, Z. Zhao, M. Backes, and Y. Zhang, "Membership inference attacks by exploiting loss trajectory," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.

[14] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer, "Detecting pretraining data from large language models," in *International Conference on Learning Representations (ICLR)*, 2024.

[15] J. Mattern, F. Mireshghallah, Z. Jin, B. Schoelkopf, M. Sachan, and T. Berg-Kirkpatrick, "Membership inference attacks against language models via neighbourhood comparison," in *Findings of the Association for Computational Linguistics (ACL)*, 2023.

[16] M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, and H. Hajishirzi, "Do membership inference attacks work on large language models?" in *Conference on Language Modeling (CoLM)*, 2024.

[17] F. P. Krüger, J. Östman, L. Mervin, I. V. Tetko, and O. Engkvist, "Publishing neural networks in drug discovery might compromise training data privacy," *Journal of Cheminformatics*, 2025.

[18] M. Lassila, J. Östman, K.-H. Ngo *et al.*, "Practical Bayes-optimal membership inference attacks," in *Annual Conference on Advances in neural information processing systems (NeurIPS)*, 2025.

[19] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Knock knock, who's there? membership inference on aggregate location data," in *Network and Distributed Systems Security (NDSS) Symposium*, 2018.

[20] A. Voyez, T. Allard, G. Avoine, P. Cauchois, E. Fromont, and M. Simonin, "Membership inference attacks on aggregated time series with linear programming," in *International Conference on Security and Cryptography (SECRYPT)*, 2022.

[21] F. Mireshghallah, K. Goyal, A. Uniyal, T. Berg-Kirkpatrick, and R. Shokri, "Quantifying privacy risks of masked language models using membership inference attacks," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[22] M. Meeus, S. Jain, M. Rei, and Y.-A. de Montjoye, "Did the neurons read your book? document-level membership inference for large language models," in *USENIX Security Symposium (USENIX Security)*, 2024.

[23] M. Meeus, I. Shilov, S. Jain, M. Faysse, M. Rei, and Y.-A. de Montjoye, "SoK: Membership Inference Attacks on LLMs are Rushing Nowhere (and How to Fix It)," in *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025.

[24] N. Koren, A. Goldsteen, G. Amit, and A. Farkash, "Membership inference attacks against time-series models," in *Proceedings of the Asian Conference on Machine Learning (ACML)*, 2024.

[25] L. Rossi, M. Aerni, J. Zhang, and F. Tramèr, "Membership inference attacks on sequence models," in *IEEE Security and Privacy Workshops (SPW)*, 2025.

[26] H. Li, Z. Li, S. Wu, C. Hu, Y. Ye, M. Zhang, D. Feng, and Y. Zhang, "Seqmia: Sequential-metric based membership inference attack," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024.

[27] T. Chen, A. Asai, N. Mireshghallah, S. Min, J. Grimmelmann, Y. Choi, H. Hajishirzi, L. Zettlemoyer, and P. W. Koh, "CopyBench: Measuring literal and non-literal reproduction of copyright-protected text in language model generation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

[28] N. Kandpal, K. Pillutla, A. Oprea, P. Kairouz, C. A. Choquette-Choo, and Z. Xu, "User inference attacks on large language models," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

[29] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *International Journal of Forecasting*, vol. 9, no. 4, 1993.

[30] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, "Ts2vec: Towards universal representation of time series," in *AAAI conference on artificial intelligence*, vol. 36, no. 8, 2022.

[31] A. Jalaly Bidgoly, H. Jalaly Bidgoly, and Z. Arezoumand, "A survey on methods and challenges in eeg based authentication," *Computers & Security*, vol. 93, 2020.

[32] A. Voyez, T. Allard, G. Avoine, P. Cauchois, E. Fromont, and M. Simonin, "Unique in the smart grid -the privacy cost of fine-grained electrical consumption data," *Scientific Reports*, vol. 15, no. 17391, 2025.

[33] I. Obeid and J. Picone, "The temple university hospital eeg data corpus," *Frontiers in Neuroscience*, vol. 10, 2016.

[34] scikit learn, "sklearn.preprocessing.robustscaler." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

[35] A. Trindade, "ElectricityLoadDiagrams20112014," UCI Machine Learning Repository, 2015, DOI: https://doi.org/10.24432/C58C86.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, 1997.

[37] C. Challu, K. G. Olivares, B. N. Oreshkin, F. Garza Ramirez, M. Mergenthaler Canseco, and A. Dubrawski, "Nhits: Neural hierarchical interpolation for time series forecasting," *AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023.

[38] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," in *International Conference on Learning Representations (ICLR)*, 2020.

[39] C. Challu, K. G. Olivares, B. N. Oreshkin, F. Garza, M. Mergenthaler-Canseco, and A. Dubrawski, "N-HiTS: Neural hierarchical interpolation for time-series forecasting," https://github.com/cchallu/n-hits, 2023, gitHub repository.

[40] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[41] S. Shachor, N. Razinkov, and A. Goldsteen, "Improved membership inference attacks against language classification models," *arXiv 2310.07219 [cs.LG]*, 2024.

[42] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, 2020.

[43] A. Ismail-Fawaz, M. Devanne, S. Berretti, J. Weber, and G. Forestier, "Look into the lite in deep learning for time series classification," *International Journal of Data Science and Analytics*, 2025.

[44] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The ucr time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, 2019.

[45] J. Hayes, I. Shumailov, C. A. Choquette-Choo, M. Jagielski, G. Kaissis, K. Lee, M. Nasr, S. Ghalebikesabi, N. Mireshghallah, M. S. M. S. Annamalai, I. Shilov, M. Meeus, Y.-A. de Montjoye, F. Boenisch, A. Dziedzic, and A. F. Cooper, "Exploring the limits of strong membership inference attacks on large language models," in *Annual Conference on Advances in neural information processing systems (NeurIPS)*, 2025.

[46] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016.

APPENDIX

A. ROC Curves

In this appendix, we report the ROC curves corresponding to the results presented in Sec. VII for both the online and the offline setting.

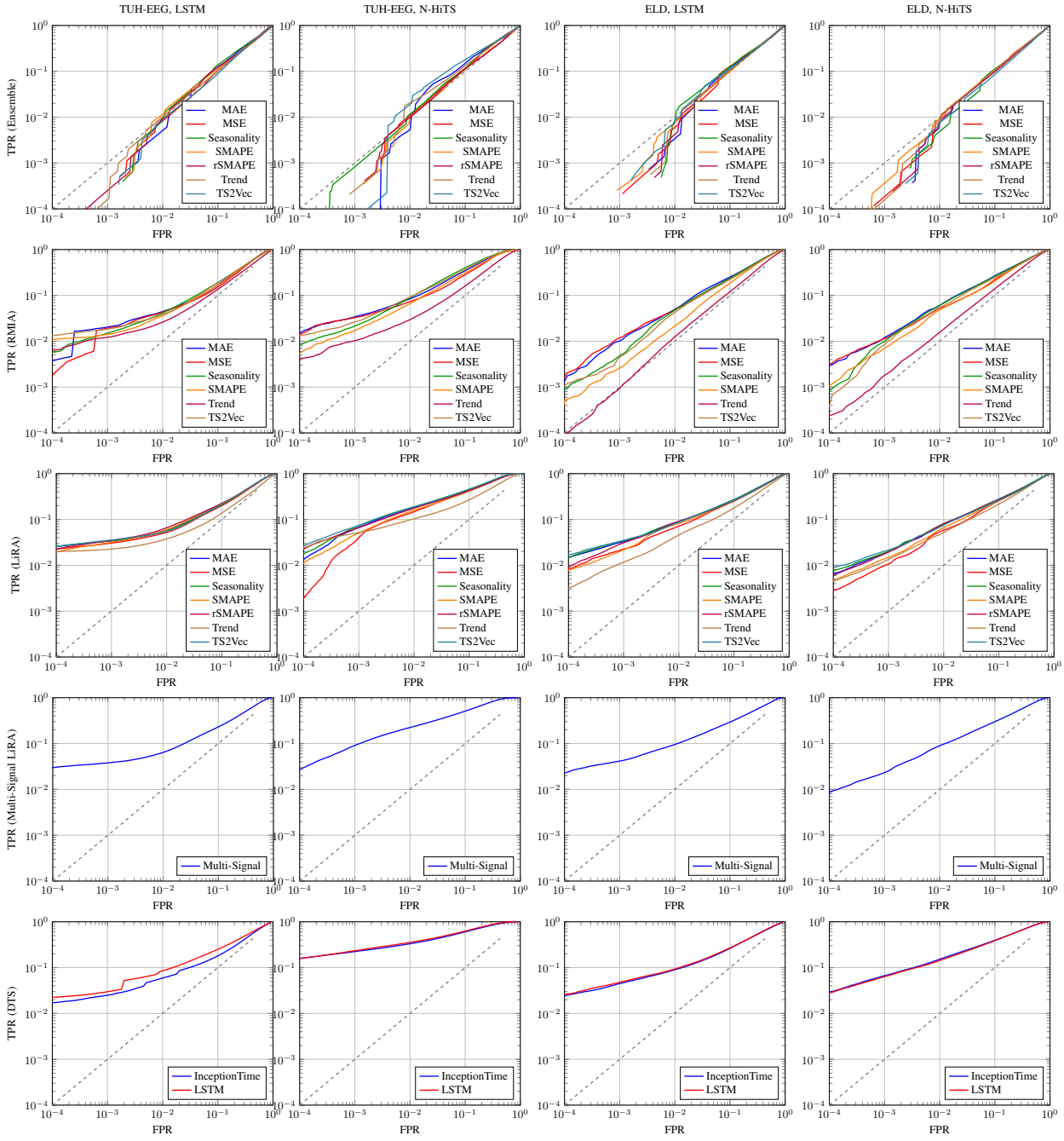


Fig. 4: ROC curves with logarithmic scales (TPR on the y-axis and FPR on the x-axis) for five membership inference attacks: Ensemble, RMIA, LiRA, Multi-Signal LiRA, and DTS, arranged in rows. Columns correspond to dataset–architecture pairs: EEG–LSTM, EEG–N-HiTS, ELD–LSTM, and ELD–N-HiTS. All results are for the **online** setting, with the Ensemble attack evaluated in audit mode. Within each subplot, curves correspond to individual attack signals for LiRA, RMIA, and Ensemble, all combined signals for Multi-Signal LiRA, and different classifier architectures for DTS. Each curve represents the mean over five independent runs.

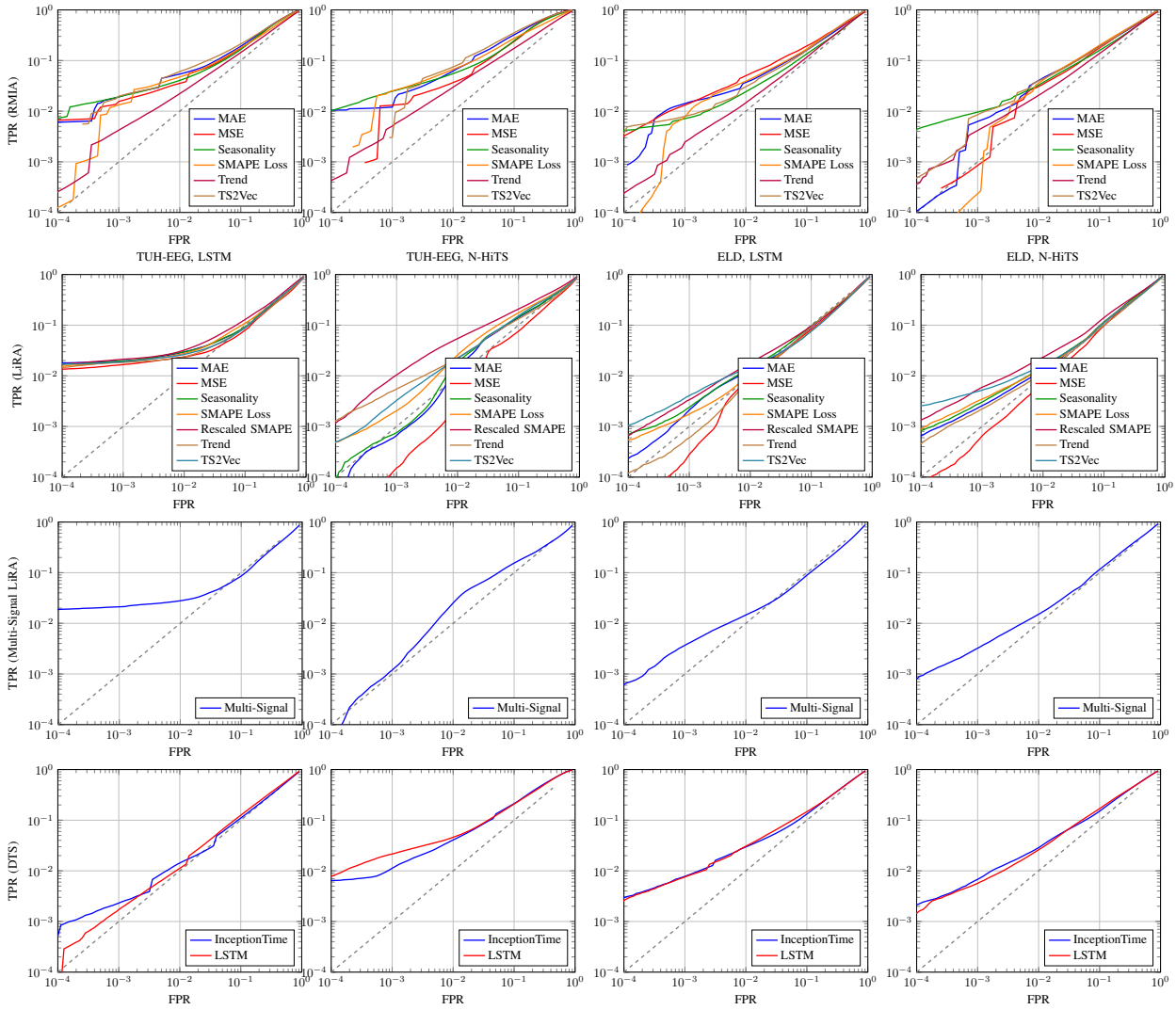


Fig. 5: ROC curves with logarithmic scales (TPR on the y-axis and FPR on the x-axis) for four memberships inference attacks: RMIA, LiRA, Multi-Signal LiRA, and DTS, arranged in columns. Rows correspond to dataset–architecture pairs: EEG–LSTM, EEG–N-HiTS, ELD–LSTM, and ELD–N-HiTS. All results are for the **offline** setting. Within each subplot, curves correspond to individual attack signals for LiRA and RMIA, all combined signals for Multi-Signal LiRA, and different classifier architectures for DTS. Each curve represents the mean over five independent runs.

TABLE V: Target-model losses for N-HiTS trained on ELD under DP-SGD with clipping $C = 10$, batch size 1024, and noise multipliers $\sigma \in \{0, 0.2, 0.8, 1\}$, along with DTS-online and Multi-Signal LiRA-Online attack performance on the mentioned models at 0.1% FPR. The model losses and attack performance for the non–DP-SGD baseline ($\sigma = 0$, and $C =$ None) are also reported for reference.

σ	C	DTS (InceptionTime) 0.1% FPR	Multi-Signal LiRA 0.1% FPR	Losses MSE	MAE	SMAPE	ND
0.0	None	7.12	3.32	0.034314	0.088645	0.166489	0.163490
0.0	10	5.69	3.02	0.344160	0.091787	0.178334	0.177658
0.2	10	5.71	2.85	0.034814	0.094275	0.175760	0.173873
0.8	10	4.91	1.91	0.037147	0.098007	0.181079	0.180756
1.0	10	3.26	1.88	0.039224	0.105430	0.195587	0.195476

B. Mitigation via DP-SGD

Membership inference is fundamentally tied to memorization, leading to systematic differences between predictions on members and non-members [7], [12]. Some mitigation strategies therefore aim to reduce such memorization, for example through regularization techniques such as dropout, weight decay, and early stopping. In all main experiments in Section VII, we employ early stopping as part of the standard training procedure.

Beyond regularization-based methods, differential privacy is commonly adopted to provide further resilience. In particular, DP-SGD [46] injects calibrated noise during training and offers formal privacy guarantees.

To examine this effect in forecasting models, we conduct an experiment where an N-HiTS target model is trained on the ELD dataset using DP-SGD with the clipping norm fixed at $C = 10$, batch size 1024, and the noise multiplier varied as $\sigma \in \{0.0, 0.2, 0.8, 1.0\}$. We then attack the target model under the record-level threat model using DTS and Multi-Signal LiRA in the online setting.

Table V reports attack performance and target-model losses (MSE, MAE, SMAPE, ND) across different noise levels. Results without DP-SGD ($\sigma = 0$ and $C =$ None) are also included. As expected, more noise consistently weakens the MIA while deteriorating the losses. In particular, the performance of the DTS attack under the most private setting ($\sigma = 1.0$) yields TPR of 3.26% at an FPR of 0.1%. This is less than half the TPR observed without DP-SGD.