# A Representation Learning Induced Property Inference Attack on Machine Learning Models for E-Health

Moomal Bukhari†, Asif Iqbal‡, Muhammad Naveed Aman†, *Senior Member, IEEE*, and Biplab Sikdar‡, *Senior Member, IEEE*
†School of Computing, University of Nebraska-Lincoln, Lincoln, NE 68588 USA
‡Department of Electrical and Computer Engineering, National University of Singapore, Singapore.
Emails: mbukhari2@unl.edu, aiqbal@nus.edu.sg, naveed.aman@unl.edu, bsikdar@nus.edu.sg

***Abstract*—Privacy concerns have become increasingly prominent as machine learning (ML) models are adopted in an increasing number of sectors. The potential of unintended or malicious exposure of sensitive data, especially in E-Health solutions, has increased as these models are shared and deployed more broadly. In order to highlight the important problem of property inference attacks, which can result in privacy and data confidentiality breaches, this study focuses on inferring global characteristics of the underlying datasets used to train the ML models. Building upon the intriguing work by Ateniese *et al.* on property inference attacks on ML models, we present a novel property inference attack using Variational Auto-Encoders (VAEs). VAEs offer a strong answer to the difficult problem of inferring dataset attributes because of their reputation for being successful in modeling complex data distributions and producing synthetic data samples. Experiments on three healthcare and the US census datasets show that the proposed attack can effectively reveal underlying patterns in the training dataset with up to 94.29% accuracy. A comparison with the popular meta-classifier based property inference attacks shows that the proposed attack not only has better success rate, but can do so with half training data and a smaller number of shadow models.**

***Index Terms*—property inference, e-health, neural networks, variational auto-encoders.**

## I. Introduction

Machine learning (ML) has emerged as a revolutionary force in a variety of sectors, highlighting its enormous potential. It continues to transform sectors including but not limited to healthcare, banking, and transportation by allowing computers to learn from data and make better predictions or judgments without the need for explicit programming. Big data is essential to identify patterns, generate precise predictions, and enhance overall performance of ML models. This has thus sparked a tendency in the industry to share ML models to foster teamwork, quicken the pace of innovation, and make advanced artificial intelligence (AI) capabilities more accessible to all. The unrestricted sharing of models, however, has prompted serious privacy concerns. This is particularly critical for models trained on electronic health records (EHR), as they contain individuals' medical history. Leakage of such information can be more costly than other types, such as business, banking, and government data [1]. Tight security measures and ethical considerations are gaining importance in the ML community since unauthorized access to private data or the improper usage of shared models may undermine data privacy and lead to dire consequences. However, striking a balance between the advantages of open cooperation and the data privacy preservation is a challenging task.

Patient privacy violations in machine learning models trained on EHRs can occur when sensitive patient information is inadvertently disclosed or inferred from the model's predictions. This way, ML models may reveal patients' identities or sensitive health conditions, even if the training data was anonymized. For instance, predictive models could expose patients' age, gender, geographic location, or medical history, potentially leading to re-identification. To address these risks, robust privacy-preserving techniques such as differential privacy and data anonymization could be employed [2], along with adherence to strict data governance policies and regulatory requirements like HIPAA. These measures are essential to safeguard patient confidentiality and ensure responsible data handling practices throughout the ML lifecycle.

There are many different types of privacy attacks on ML models of which the two most common types are model inversion and membership inference attacks [3]. By attempting to ascertain if a particular data point was included in a model's training dataset, membership inference attacks have the potential to violate data privacy [4]. Conversely, model inversion attacks probe a model's output in an attempt to retrieve private information about a specific person. Although both of these attacks target individual records, property inference attacks are more focused on inferring sensitive global features of a dataset, such as the prevalence of specific traits within the data. Due to the possibility of disclosing private information about the inner workings of ML models, property inference attacks carry a high risk of serious security implications. For example, for a malware classifier trained using execution traces, an adversary can use property inference to identify the characteristics of the testing environment, which can reveal vulnerabilities and

This research is supported by A*STAR, CISCO Systems (USA) Pte. Ltd and National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002).

evasion strategies [5]. Similarly, property inference can reveal weaknesses in an email filtering system by having a spam classifier take into account attributes like average sentiment or keyword presence [6]. An adversary can reveal the model's biases using targeted queries. Then, they can modify email content to get past spam filters, jeopardizing the security of email systems and flooding users with dangerous content.

Recent research has revealed disparities in the representation of particular demographic groups in various training datasets, including minorities and women. Variations in the efficacy of standard classifiers among different groups have been attributed to this representational bias [7]. Investigating whether the dataset used for model training exhibits a higher or lower representation of a particular class, which represents a broader characteristic of the dataset, is therefore a common interest.

The significance of fighting against property inference attacks stems from the potentially disastrous effects of disclosing broader, population-level knowledge. It is imperative to defend against such attacks in order to preserve the privacy and confidentiality of sensitive data, particularly in industries such as healthcare and finance, where even aggregate data might be misused for nefarious intent or biased profiling. Thus, it is important to identify all possible property inference attacks. Differential privacy appears to provide limited or no protection against property inference attacks, as it was designed primarily to provide privacy assurances in instances involving membership inference attacks [8]. Relatively little study has been done on property inference attacks and the defenses that go along with them, which is noteworthy and indicates a major need in the field [3].

Ateniese et al. were among the first to study property inference attacks [8]. The attack involved training a meta-classifier to identify whether a target classifier has a particular property, $P$, or not. The adversary creates a collection of substitute classifiers, which are commonly known as shadow classifiers. Each of these shadow classifiers is trained on a dataset that has been purposefully designed to either possess or lack the property $P$. They are trained for the same task as the target classifier. The parameters of these shadow classifiers are then used to train the meta classifier. In addition to demonstrating how this attack can be used against classifiers such as Support Vector Machines (SVMs) and Hidden Markov Models (HMMs), Ateniese et al. also pointed out that this type of property inference cannot be effectively thwarted by the use of differential privacy mechanisms that are intended to provide record-level privacy.

When trained with unlabeled data, the representation learning based variational auto-encoders (VAEs) demonstrate exceptional performance [9]. Through the seamless integration of auto-encoders and probabilistic modeling, VAEs produce continuous, organized latent space representations. They stand out from other generative models like conventional auto-encoders and generative adversarial networks (GANs) due to their probabilistic character. This makes them useful for applications where uncertainty is a concern, such as natural language processing and image generation. Due to their ability to learn a continuous, ordered representation of the data, VAEs allow for flexible manipulation in the latent space.

In this paper, we propose a property inference attack framework using the VAE. By harnessing the capability of VAEs to generate samples from the same distribution as the training data, we exploit their ability to distinguish samples lying outside the training distribution. We evaluate our approach on four datasets, including the default baseline US Census dataset and three healthcare-related datasets. Our experimental results demonstrate the superiority of the proposed VAE-based attack framework over existing meta-classifier-based techniques. This paper presents a novel method of using VAEs for property inference attacks with the following major contributions:

1) A VAE based property inference attack on ML models.
2) A thorough evaluation of the proposed attack on four different datasets.
3) A comparison with the existing state-of-the-art property inference attacks.

This paper is organized as follows: Section II explains the threat model and discusses the underlying assumptions. In Section III we formulate the problem and in Section IV, we present the proposed property inference attack. Experiment setup along with the results and findings are presented in Section V, which provides a thorough analysis of the conclusions drawn from our experimentation. We conclude the paper in Section VI.

## II. Threat Model

Suppose that a fully connected neural network (FCNN) $h$ is trained for classification using a training dataset $L$. Following training, this model is shared with consumers, allowing them to make predictions. The objective of the adversary is to identify training data properties when they are only provided access to the model $h$.

### A. Assumptions

We make the following two key assumptions. Firstly, we presume that the adversary has white box access of the model and thus, has knowledge of the model architecture and parameters [5]. This is a fair assumption given that models are openly shared on many different platforms. Additionally, there exist methods for effectively extracting the models when the user is just given an executable software file for predictions, or when users have the option to use ML models as a service through an Application Programming Interface (API) [10, 11]. Secondly, we assume that the adversary cannot alter the data collection processes that compose the target model's training dataset or tamper with the target model's training process. This suggests that this study does not take integrity assaults into consideration, hence eliminating the chance that an opponent could secretly impart the target property during training by surreptitiously embedding covert information into the model. These presumptions set the parameters for the study's investigation of property inference, guaranteeing a problem-focused investigation [12].

## III. Problem Formulation

FCNNs are the main building blocks in ML tasks due to their adaptability. They are made up of multiple layers of networked computational units, or neurons, where each one processes data using a weighted sum followed by an activation function. The output of a single neuron, $o$ is computed as:

$$o = \gamma(Wx + b), \tag{1}$$

where $W$ is the weight vector, $x$ is the input data vector, $b$ is the bias term, and $\gamma$ is the non-linear activation function (`sigmoid` or `ReLU`).

The following expression represents the output of the $i^{th}$ neuron in the $t^{th}$ layer in a fully connected layer:

$$o_i^t = \gamma(w_i^t.o^{t-1} + b_i^t), \tag{2}$$

where $o^{t-1}$ is the output of the previous layer, $w_i^t$ is the weight vector of the $i^{th}$ neuron in the $t^{th}$ layer, and $b_i^t$ is the respective bias vector. Computation of the output of the network, denoted by $y$, from a given input $x$, involves a sequence of transformations through the layers of the FCNN, given as:

$$y = h(x) = H_{|h|}(H_{|h|-1}(\dots(H_2(H_1(x))))), \tag{3}$$

where $h$ represents the neural network function, $|h|$ is the number of computational layers (hidden and output layers), and $H_i$ is the transformation function in the $i^{th}$ layer. The transformation functions are contingent on the nature of the task. For example, recurrent layers are used for text analysis, whereas convolutional layers are utilized for image processing. Hyperparameters are terms used to describe training-related variables such as learning rate and network structure such as the number of layers and neurons in each layer.

Utilizing backpropagation and optimization techniques, the network attempts to minimize a specified loss function, such as mean squared error for regression or binary cross-entropy for classification tasks. By updating the network's weights and biases repeatedly, this iterative approach improves its prediction performance. The aforementioned equations summarize the fundamental ideas and procedures that support neural networks, which are now widely used in ML applications. In summary, input data $x$ is mapped to the output $y$ using a neural network as follows:

$$y = h(x, W_{NN}, B), \tag{4}$$

where $W_{NN}$ is the model weights of the trained neural network and $B$ is the bias vector.

VAEs are a form of generative model commonly employed in unsupervised learning. Because VAEs use probabilistic reasoning, they can create new data points by sampling from the learnt latent space distribution, which sets them apart from other algorithms. An encoder and a decoder make up the two basic parts of a VAE. The main innovation of VAEs is the way they learn the latent space's probability distribution using variational inference, which enables sampling and the creation of new data points. The encoder is trained to map the input samples into a smooth and connected latent space (trained to be similar to the standard Normal distribution), and samples from the latent space are mapped back to the data space by the decoder. The method of training a VAE entails optimizing the Evidence Lower Bound (ELBO) by modifying the encoder and decoder parameters.

VAEs are trained using the ELBO as the objective function. The reconstruction term and the regularization term are its two constituent terms. It is mathematically defined as:

$$ELBO = E[logP(X|z)] - KL(q(z|X)||p(z)) \tag{5}$$

where $E$ represents the expectation, $X$ represents the input data, $P(X|z)$ is the log-likelihood of the data given the latent space representation (modeled by the decoder network), $q(z|X)$ is the estimated posterior distribution of the latent space (modeled by the encoder network), $p(z)$ is the prior distribution in the latent space, and $KL$ is the Kullback-Leibler divergence. ELBO acts as a trade-off between the latent space's regularization (KL divergence term) and reconstruction quality (log-likelihood term). VAEs seek to produce data that closely resembles the input while guaranteeing that the latent space adheres to a particular prior distribution by maximizing the ELBO. The latent variable $z$, which is the input to the decoder network is sampled from the latent space parameterized by the encoder using the reparameterization trick, shown as:

$$z = \mu + \sigma * \epsilon \tag{6}$$

where $\epsilon \sim \mathcal{N}(0,\ 1)$, $\mu$ and $\sigma$ are the outputs of the encoder network. The reparameterization trick ensure the end-to-end differentiability of the VAE network.

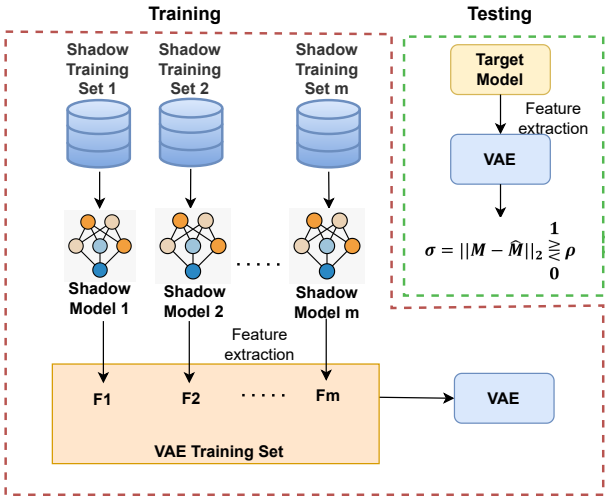



Fig. 1: Proposed property inference attack.

## IV. Proposed Property Inference Attack

The proposed property inference attack framework is shown in Fig. 1, where, the features extracted from target model are passed through the trained VAE model, and its reconstruction error is used to hypothesize whether the data used to train

the target model contained the target property or not. Let $h_{target}$ be the target model, and $(h_1, \ldots, h_m)$ be the $m$ shadow models. It is critical to underscore that within our approach, the shadow models undergo training exclusively on datasets that contain or lack the property $P$. Clarifying the meaning of the term "property" in relation to a dataset is of the utmost importance. For instance in a healthcare dataset, $P_1$ may denote the presence of a particular disease in the patient's data record, and $P_2$ could indicate patient's gender. On the other hand, $P_1$ may suggest a high probability of sunny days in the context of weather forecast, but $P_2$ may suggest an equal chance of all weather conditions. The aforementioned examples serve to demonstrate that $P_1$ and $P_2$ are not negations of one another, but rather symbolize alternative options, and the attacker is free to select any type of property. In this work, we study binary property inference. For the sake of simplicity, we therefore always refer to $P_1$ as $P$ and $P_2$ as $\bar{P}$.

The attack is carried out as follows:

1) Shadow training sets are created that contain the property of interest $P$.
2) Training the shadow models, which are implemented using neural networks, using the shadow training sets.
3) The model weights of each individual shadow classifier are taken as the training features, represented as $F_1, F_2, \ldots, F_m$. We then use these feature representations to build a large-scale dataset for training the VAE.

Specifically designed for the purpose of anomaly identification, this VAE uses the shadow model weights to identify and recognize aberrant patterns. Using the shadow models neural network weights as our training dataset for the VAE training, the VAE is trained to detect the presence of a property in the target model. The idea is that a trained VAE will be able to reconstruct the network weights of a model trained on dataset containing the property $P$ well, and would lead to high reconstruction errors otherwise. To make this classification, the VAE's reconstruction error, $\sigma$, is compared with a threshold, $\rho$. If the reconstruction error is greater than the threshold the anomaly is detected, represented by 1, indicating that the property $P$ is not present. If $\sigma$ is less than the threshold, the anomaly is not detected, represented by 0, indicating that the target model has the property $P$. The testing block in Fig. 1 depicts this classification with the reconstruction error given by $\sigma = ||M - \hat{M}||_2$, where $M$ is the input model weights and $\hat{M}$ is the reconstructed model weights by the VAE.

Property inference leverages the assumption that ML models that have been trained using comparable data and methodologies will demonstrate similar underlying functions. These common patterns in their parameters represent these shared functions. Adversary's goal is finding these patterns in the target model in order to reveal particular characteristics that the model's designer might not have wanted to reveal.

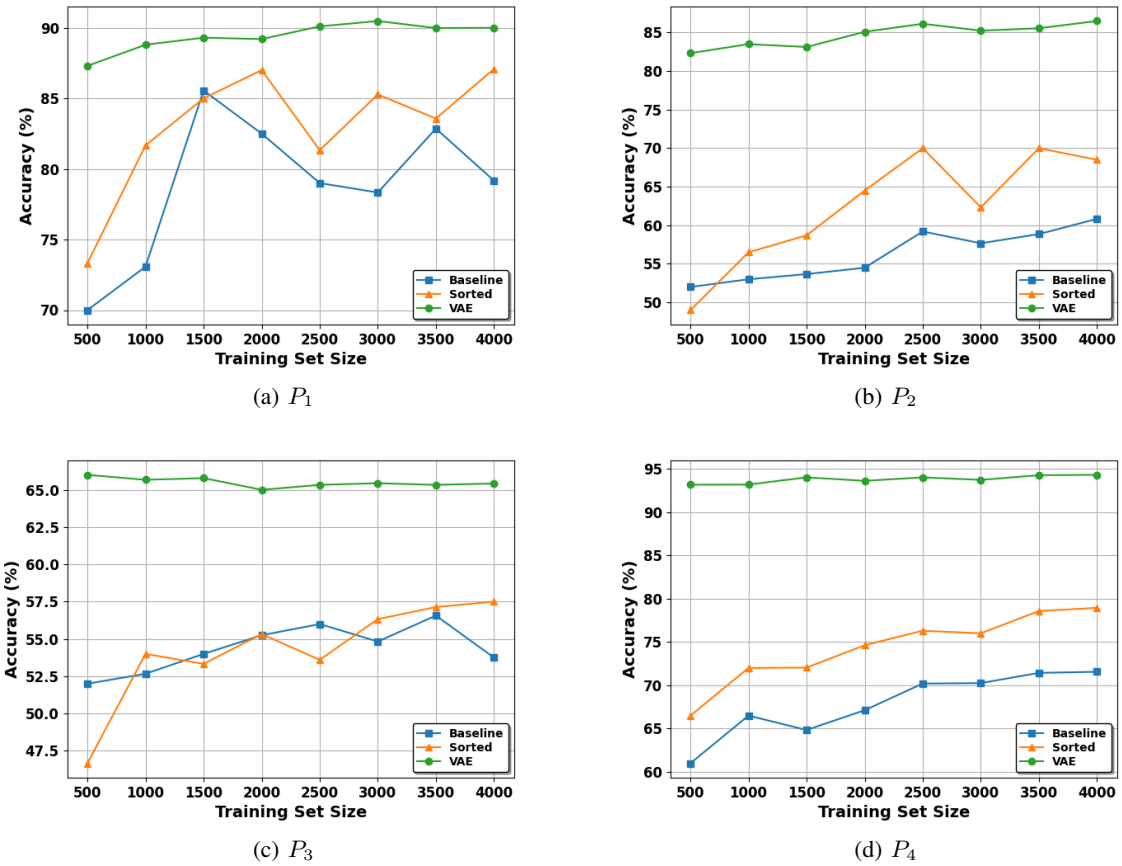
## V. Experimental Evaluation

We analyzed the proposed property inference attack using a unique approach, leveraging four datasets to evaluate our technique. The first dataset is the US Census dataset [13] which is a collection of data gathered from demographic surveys carried out in 1994 and 1995 by the U.S. Census Bureau. It has 299,285 records, each of which has 41 characteristics pertaining to employment and demographics, including citizenship, marital status, education, occupation, and race/gender. This dataset is included in the analysis as it serves as a default dataset commonly used in many property inference attack works. The main goal is to use the available census attributes to estimate if a person makes more than $50,000 per year or not. The dataset was preprocessed and cleaned for missing values using OpenRefine [14].

The second dataset we utilized pertains to cardiovascular disease, the Framingham Heart Disease dataset (FHS) [15]. It is worth noting that the original dataset utilized in our analysis had missing values and a "education" characteristic was eliminated due to its insignificance to the classification job at hand. Preprocessing was done on the dataset to fix these missing values and guarantee data consistency using mean value of the attributes to fill the missing values for continuous valued attributes whereas for binary valued attributes with missing values, the records were discarded from the training set. Fourteen different features such as gender, age, smoking status, etc are included in this extensive dataset. In addition, the dataset has a sixteenth feature that shows the chance of getting coronary heart disease (CHD) over a ten-year period. These traits are used to forecast if a patient has a 10-year chance of developing CHD in the future. The third dataset we used is Sepsis Survival Minimal Clinical Records datasets [16]. The dataset includes medical records from 110,204 admissions in Norway between 2011 and 2012 that involved 84,811 hospitalized patients. The aim is to forecast the survival status of the patient roughly nine days post-hospitalization by utilizing three characteristics: age, gender, and the number of episodes. The fourth dataset we worked on is CDC Diabetes Health Indicators [17]. The dataset has 35 features, including some demographic information for each individual such as age, gender, and so on, while the remainder of the features are health-related, such as if the person has high blood pressure or is a smoker, among others.

In the evaluation we compare our technique with the baseline approach proposed in [5] which is based on the state-of-the-art property inference attack by Ateniese et al. [8] and the neuron sorting approach [5]. The baseline strategy trains the meta-classifier directly on the raw shadow classifier parameters, while the sorting approach sorts the parameters by the sum of weights in descending order [5]. First, for every dataset we create a fleet of $4,000$ neural network-based shadow models with property $P$ for training the VAE. For the baseline strategy and the sorting strategy, for every dataset we create a fleet of $4,000$ neural network-based shadow models out of which half of them are with property $P$ and other half are with property $\bar{P}$.

In order to train models using the US Census dataset, neural networks with three hidden layers of 32, 16, and 8 sizes are utilized. For training the models using Framingham




(a) $P_1$ (b) $P_2$

(c) $P_3$ (d) $P_4$

Fig. 2: Accuracy of attacks.

TABLE I: Experiment Details

| **Experiment** | **Dataset** | **Target Property** $P$ |
|---|---|---|
| $P_1$ | US Census | Higher proportion of women (80% of Women) |
| $P_2$ | Framingham Heart Disease | Increased total cholesterol levels greater than 240 mg/dL (80%) |
| $P_3$ | Sepsis Survival | Higher proportion of women (85% of Women) |
| $P_4$ | CDC Diabetes | Higher proportion of women (80% of Women) |

dataset neural networks two hidden layers of 32 and 16 sizes are utilized. For models trained on Sepsis dataset and CDC diabetes dataset, neural networks having two hidden layers of sizes 16 and 8 are used. PyTorch is used to train every neural network model. We utilize the Adam optimizer with 40 maximum training epochs, ReLu as the activation function, a learning rate of 0.001, and a weight decay of 0.01 for all of our training models. The purpose of the meta classifiers that are set up for binary classification to determine whether the property $P$ is present in the training dataset of the target. The activation function used for all hidden layers is `ReLU`, whereas `sigmoid` activation functions is used for the output layer of the meta classifier.

In Table I, we display the properties targeted for deduction from the target model. To determine the threshold $\rho$ required by our proposed model, we set a false positive rate (FPR) over the training dataset. Specifically, with a trained VAE model, we computed the reconstruction error of the entire training set and selected the error value as $\rho$ corresponding to the desired FPR. We conducted all experiments 10 times, each with a fresh permutation of the training dataset. Additionally, the target model was retrained for each trial. The resulting accuracies reported in Fig. 2 represent the average across these trials. The FPR used for the VAE model was set to 10%. Observing the figure, we notice that the sorted and baseline methods exhibit improved attack accuracy as the size of the training set increases. Conversely, the proposed attack using the VAE shows consistent accuracy across different training set sizes.

TABLE II: Property inference attack accuracy under multiple false positive rates.

| **FPR** $\rightarrow$ | 10% | 5% | 1% |
|---|---|---|---|
| $P_1$ | 90 | 89.56 | 77.33 |
| $P_2$ | 86.50 | 82.5 | 71.37 |
| $P_3$ | 65.42 | 65.00 | 65.10 |
| $P_4$ | 94.29 | 95.75 | 89.25 |

This suggests that the proposed method achieves higher attack success rates with relatively smaller training datasets, requiring fewer shadow models to be trained.

To further analyze the proposed attack's accuracy across various FPR values, we repeated the experiment for all four datasets using FPR values of 10%, 5%, and 1% for a fixed training set size of $4,000$. The results are presented in Table II, indicating that the attack success rate remains stable under FPRs of 10-5%, but decreases significantly under 1% FPR. Nevertheless, even at 1% FPR, the proposed model achieves higher accuracy than the baseline and sorted methods for $P_2$, $P_3$, and $P_4$. Moreover, for FPR of 5% and $P_1$, the proposed model's accuracy surpasses that of the compared methods. These observations underscore the robustness of the proposed attack model to threshold selection.

In light of the meta classifiers' data requirements, the reduced accuracy is therefore well within the anticipated range. This result highlights the potential benefit of our methodology, which uses VAE to perform property inference attacks with comparatively lower data requirements, while maintaining a respectable level of property $P$ identification accuracy. When the training set size is the same, it is clear from a direct comparison that our proposed attack performs significantly better than the attack proposed in [5]. This significant gap draws attention to a critical advantage that the proposed attack provides when it comes to property inference attacks, i.e., the proposed attack does not require labelled data for training where labeled data may be hard to come by or difficult to access.

## VI. Conclusion

This paper offered light on the under explored subject of property inference attacks, a vital part of data security in the context of machine learning. The proposed method, which makes use of VAEs trained with the weights of shadow models can effectively be used for determining if global attributes are present in a target model or not. Experiments on actual data showed that the proposed attack can reveal underlying data properties of a machine learning model with upto 94.29% accuracy. Moreover, a comparison with the state-of-the-art showed that the proposed attack does not only have significantly higher accuracy but does so with approximately 50% lesser data. In summary, by demonstrating the viability of property inference attacks and the efficiency of VAEs in this situation, our research advances the field of data security. This approach opens the door for further research into improving the security of ML models while also deepening our understanding of data security.

## References


[1] R. Nowrozy, K. Ahmed, A. Kayes, H. Wang, and T. R. McIntosh, "Privacy preservation of electronic health records in the modern era: A systematic survey," *ACM Computing Surveys*, 2024.

[2] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[3] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *ACM Computing Surveys*, 2020.

[4] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18, IEEE, 2017.

[5] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pp. 619–633, 2018.

[6] S. Mahloujifar, E. Ghosh, and M. Chase, "Property inference from poisoning," in *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1120–1137, IEEE, 2022.

[7] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on fairness, accountability and transparency*, pp. 77–91, PMLR, 2018.

[8] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *International Journal of Security and Networks*, vol. 10, no. 3, pp. 137–150, 2015.

[9] S. Sinha and A. B. Dieng, "Consistency regularization for variational auto-encoders," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12943–12954, 2021.

[10] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction {APIs}," in *25th USENIX security symposium (USENIX Security 16)*, pp. 601–618, 2016.

[11] Z. Lin, X. Zhang, and D. Xu, "Automatic reverse engineering of data structures from binary execution," in *Proceedings of the 11th Annual Information Security Symposium*, pp. 1–1, 2010.

[12] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, pp. 587–601, 2017.

[13] R. Kohavi and B. Becker, "Census income (kdd) dataset." https://archive.ics.uci.edu/dataset/117/census+income+kdd. Accessed on 12-01-2023.

[14] OpenRefine Development Team, "Openrefine." https://openrefine.org/. Accessed on 12-01-2023.

[15] "Framingham_heart_disease." https://www.kaggle.com/datasets/ppranayk/framingham-heart-disease. Accessed on 09-02-2023.

[16] "Sepsis survival minimal clinical records." https://archive.ics.uci.edu/dataset/827/sepsis+survival+minimal+clinical+records. Accessed on 01-04-2024.

[17] "Cdc diabetes health indicators." https://archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators. Accessed on 02-03-2024.