October 2023

ENHANCING MODEL SECURITY: LEVERAGING USER-GENERATED IDS AS EMBEDDED WATERMARKS IN MACHINE LEARNING MODELS

Alan Gatzke

Mitchell C Mosure

Andi Wilson

Ananth Racherla

Todd C Kuehnl

See next page for additional authors

Inventor(s)
Alan Gatzke, Mitchell C Mosure, Andi Wilson, Ananth Racherla, Todd C Kuehnl, and Tyler Shane McCarthy

ENHANCING MODEL SECURITY: LEVERAGING USER-GENERATED IDS AS EMBEDDED WATERMARKS IN MACHINE LEARNING MODELS

AUTHORS:
Alan Gatzke
Mitchell C Mosure
Andi Wilson
Ananth Racherla
Todd C Kuehnl
Tyler Shane McCarthy

ABSTRACT

The potential theft or unauthorized use of machine learning models developed by a company can lead to significant financial losses and damage to the company's intellectual property. While existing methods of protecting machine learning models such as encryption or access controls can be circumvented by skilled attacker, techniques presented herein involve the integration of embedded watermarks into machine learning models. Such techniques involving the integration of embedded watermarks may not only uniquely identify a model but may also include a unique user identification/identity that can make it possible to track usage of the model and detect any unauthorized use of the model. Thus, if a model is leaked, redistributed, or misused, the watermark for the model makes it possible to identify a source of the leak/misuse, allowing for better traceability and accountability.

DETAILED DESCRIPTION

Proposed herein are techniques to address the potential theft or unauthorized use of machine learning models, which can lead to significant financial losses and damage to a company's intellectual property. Many existing methods of protecting machine learning models, such as encryption or access controls, can be circumvented by skilled attackers. However, techniques presented herein involve the integration of embedded watermarks into machine learning models. These watermarks not only uniquely identify a model but can also include a unique user identification, enabling the tracking of model usage and detection of any unauthorized use. By implementing embedded watermarks with unique user identifications, the techniques proposed herein can significantly reduce the risk of

intellectual property theft and also facilitate the detection and prosecution of any unauthorized use.

In accordance with techniques of this proposal, a watermarking system can inject user personalized watermarks in a model at runtime of the model. Broadly, operations of the watermarking system may include three high-level steps involving: 1) encoding a watermark, which is a unique identifier/identity (ID) tied to a specific user, into a form that can be hidden in the cover data of a model (e.g., a sparse pattern or algorithm could be used to change the watermark information into chosen parts of a model); 2) embedding the watermark into the cover data in which chosen parts (weights or biases) could be slightly changed such that the model's performance is not impacted; and 3) recovering the unique ID tied to the specific user (e.g., for determining unauthorized use, a leak of the model) by extracting the hidden watermark via a decoding algorithm or a trained helper model. For extracting the watermark, the extraction process can facilitate finding the watermark without letting others know it's there.

Although other watermarking mechanisms exist for protection of machine learning models, there is an unmet need for a solution that utilizes a unique watermark that can be tied to each of a number of user IDs.

Figure 1, below, illustrates example details regarding example operations that may be utilized for embedding a watermark of a unique user ID into a model.

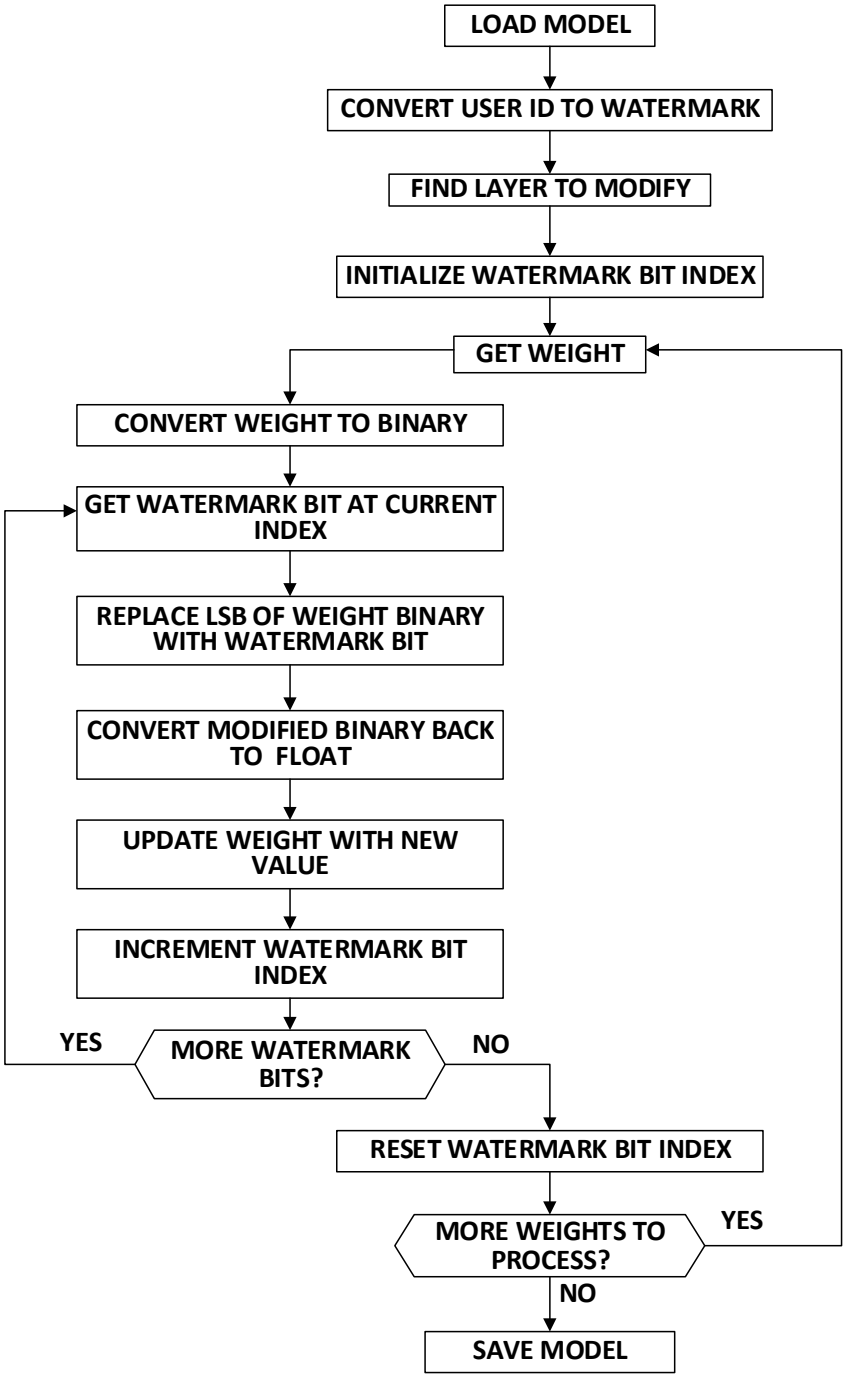

Figure 1: Example User ID Watermarking Operations

Broadly, as illustrated in Figure 1, the watermarking system can load a given model and obtain the names of the model's initializers, which can include the weights and biases that are to be modified for the model. The operations may further include converting the watermark from a float representation to a binary representation. Starting with the first bit

of the binary watermark, each weight and bias in the model can be processed such that each weight or bias value can be converted to a binary representation.

Thereafter, the least significant bit (i.e., the last bit) of the binary representation can be replaced with the current bit from the binary watermark. The modified binary representation is then converted back to its original data type (e.g., float) and the weight or bias is updated with the new value. The bit replacement process for the binary watermark can be repeated until the end of the binary watermark is reached, at which point the process can be repeated for any additional weights or biases starting again from the first bit of the binary watermark. If no more weights or biases are to be processed, the modified model is saved, and the embedding process is considered complete.

In more detail, the watermarking system provides for embedding the unique identifier in the model's compute graph by slightly altering the weights or biases of specific layers or neurons, without significantly affecting the overall model performance. This can be achieved by creating a sparse pattern of the watermark in the model's parameters or using a mathematical transformation that encodes the watermark information into the weights, while maintaining the model's functionality.

The approach may apply a steganography technique to hide the watermark within the parameters, making it challenging to detect or remove without the correct decoding algorithm. Any appropriate steganography technique may be utilized in the watermark system. For example, in some instances, Least Significant Bit (LSB) embedding may be utilized such that the least important bits of the cover data can be replaced with the bits from the watermark. Although LSB embedding may be the most straightforward technique through which to adjust the cover data on the fly without having to generate new models for each user, other techniques may be utilized in the watermark system.

For example, spread spectrum techniques may be utilized such that the watermark can be spread across the cover data using a random sequence. Other steganography techniques may include transform domain techniques in which the watermark can be embedded in the transformed domain (e.g., frequency or wavelet domain) of the cover data or Quantization Index Modulation (QIM) techniques in which a quantization process could be utilized to embed the watermark information into the model.

The novel watermarking system can also personalize the watermark by using user identifications generated from an authentication microservice. This means that each user's unique identifier is embedded in user-personalized models created on-the-fly with the watermark, allowing for the tracking and protection of the intellectual property of each individual model. Embedding the watermark in the model's compute graph involves modifying specific model parameters while maintaining functionality.

Figure 2, shown below, illustrates example details regarding example operations that may be utilized for extracting/detecting an embedded user ID for a model.

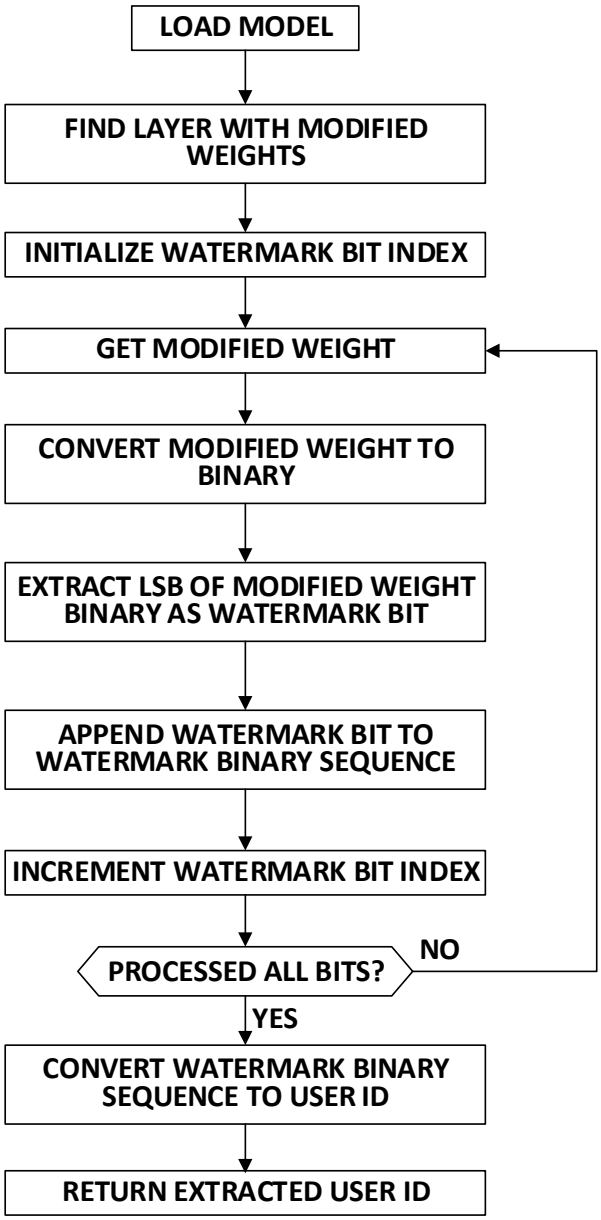

Figure 2: Example User ID Watermark Extraction/Detection Operations

Regarding Figure 2, the extraction/detection process can differ based on whether the watermark is embedded in the model weights or output media, with each method utilizing a tailored extraction technique to effectively identify the unique user identifier contained in the watermark.

For model weight embedding, the extraction/detection process involves analyzing the weights or biases of the model to extract the embedded watermark. This may involve the use of a trained auxiliary model, or a specific algorithm designed to detect the watermark in the compute graph. In contrast, when the watermark is embedded in the output media, extraction/detection process involves analyzing the output itself (e.g., images, audio, or text) to identify the unique identifier. This process may involve a separate extraction method depending on the output media type and the embedding technique used.

Accordingly, the watermarking techniques as proposed herein can be utilized to protect the intellectual property of machine learning models by embedding a unique watermark tied to a user ID. If a model is leaked, redistributed, or misused, the watermark techniques as proposed herein would make it easier to identify the source, allowing for better traceability and accountability. Knowing that there's a unique watermark embedded in a model would also discourage potential attackers or unauthorized users, since it raises the chances of getting caught and traced back to them. In case of any disputes, the watermarking system proposed herein would help verify the owner of a model or confirm that a specific user had access to the model.