Research Article

Cyberattacks Defense in Digital Music Streaming Platforms by Mobile Distributed Machine Learning

Guoxu Fan

Pingdingshan University, Conservatory of Music, Pingdingshan, Henan 467000, China

Correspondence should be addressed to Guoxu Fan; fanguoxu309@126.com

Received 29 March 2022; Accepted 19 April 2022; Published 6 May 2022

Academic Editor: Konstantinos Demertzis

Copyright © 2022 Guoxu Fan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Given the massive popularity of digital music industry repositories and their corresponding targeting by cybercriminals, this paper presents an intelligent model for cyberattacks defense in digital music streaming platforms by mobile distributed machine learning. The basic idea of machine learning is to use large data sets to create a model that responds well to inputs it has never processed before. With the increase in data volume and complexity of models, it becomes increasingly challenging to complete machine learning processes in a single machine. Distributed ML was developed to solve this problem, and a standard procedure is completed through the collaboration of multiple servers. With the evolution of mobile devices and the increase in their number, it is possible to create an integrated and compact mobile distributed machine learning (MDML) system that could reduce the workload of servers. A distributed logit polynomial function model is proposed, which is used to model options in distributed binary regression accounting units, which are of low complexity and high stability in noisy environments.

1. Introduction

One of the current challenges associated with the pandemic is the massive increase in content consumption through pay-TV and other streaming services. Users are forced to isolate themselves and have trapped many people at home [1]. From the first day of the enforcement of the measures, the music platforms available on the market and platforms for streaming content saw an increase in the number of their subscribers and the content they consume. For this reason, international streaming service providers have made recommendations for the need to reduce data rates to ensure the proper operation of networks, as the requirements have been maximized [2].

At the same time, cybercrime increased dramatically, especially the account takeover attack (ATA) [3] against music streaming platforms. Specifically, ATAs are a specialized type of attack in which criminals take ownership of online accounts using stolen access criteria to similar services. The basic techniques of these attacks are social engineering, brute force data breaches, and phishing campaigns. Botnets usually use the results of collecting these data to identify any other services that may use the same standards. In many cases, lists with similar data are sold on the dark web [4].

To further address these cases of breach and interception of credentials in a music industry service, this study proposes a highly efficient and easy-to-use MDML [5] that uses a polynomial logit function to model options in distributed binary regression units to timely detect cyberattacks in music content streaming services [6].

2. Related Literature

Mobile distributed machine learning and distributed computing [7] are concepts that the research community is currently trying to exploit in the best possible way the resources offered in each scenario [8, 9].

In 2015, Taddy [10] proposed a prototype method for the idea of distributed multinomial regression. Their research was fueled by the use of high-dimensional response multinomial models to analyze a large number of random counts. Content research, as texts are tokenized, and token counts are described as generating from a multinomial
depending on document properties, was one of its inspiring uses. They developed such algorithms using text projected onto a broad collection of explanatory factors using a publicly accessible data set of Yelp reviews. The fitted models may investigate the relationship among phrases and variables of concern, reduce dimensions into supervised component scores, and predict outcomes. We suggest that the technique presented here is an appealing choice for social engineers and other textual researchers who want to use regression tools they are acquainted with on text data.

Shamili et al. [11] developed a dispersed support vector machine technique for detecting computer viruses on a system of portable devices in the context of network security. The lightweight design uses a statistical classification system developed via training with instances of both regular and exceptional use patterns to monitor cellular user behavior in a dispersed and privacy-preserving manner. They claim that the distributed learning technique has many benefits, including being lightweight regarding Internet use, maintaining the privacy of participating users, and automatically generating a generic behavioral signature of virus based on typical user usage patterns. The system was tested employing the MIT reality data set, and the results were positive.

Gu et al. [12] conducted a literature study on the server-based to client-based machine learning transition. They again went through a few popular server-based and client-based deep learning methodologies and applications. They also spoke about the obstacles and potential future developments in this field. They have described their goals and showed how client-based machine learning is both sufficient and necessary. They have highlighted the limitations of client-based inference and illustrated recent achievements, particularly in the disciplines of machine vision and natural language interpretation. Finally, they have identified future research paths in academia and business for client-based machine learning. In conclusion, implementing client-based machine learning in real-world applications is still in progress.

Shakarami et al. [13] presented a study of ML-based computation offloading techniques in the mobile edge computing ecosystem in the shape of a classical taxonomy to identify current mechanisms and unresolved concerns in this critical field. Reinforcement training, guided learning, and uncontrolled learning were the three primary categories in their proposed taxonomy. Then, depending on various characteristics, the applicable methodologies were compared to one another. Finally, they discussed several crucial research problems as outstanding topics in the ML-based offloading mechanisms, considering the existing literature gap.

In a cognitive eavesdropping context, Guo et al. [5] examined a distributed machine learning strategy for a multiuser mobile edge computing system in a perceptual eavesdropping context, where several secondary devices have specific tasks to compute with varied priorities. They looked at the federated learning methodology for the system architecture of a multiuser mobile edge computing platform. Various users had distinct computational tasks that needed to be calculated by different computational access points. Finally, simulation results showed that the suggested strategy might successfully minimize system costs in both delay and energy usage while also ensuring that the user with the highest job priority receives greater bandwidth and processing capabilities. In future research expanding from this study, they will investigate some fairness disparities between consumers due to aspects such as the network state, job size, and processing capabilities. Furthermore, since most mobile devices have a limited battery life, it is difficult to maintain all mobile devices online.

3. Methodology

The proposed implementation, based on distributed multinomial logistic regression (DMLR) [10], is a classification method that extends the capabilities to multiclass issues, such as those with more than two possible unique outcomes [12]. In statistics, it is a model used to forecast the probability of several possible effects of a categorically distributed dependent variable, given a set of independent variables that might have varied values such as real, binary, category, and other types [5, 14].

The model follows the same technique as the accounting regression, with the sole variation being that the dependent variables are categorical rather than binary. In particular, there are \( K \) alternative outcomes rather than just two. The proposed solution uses a linear prediction function \( f(k, i) \) to predict the probability that the observation \( i \) will have an effect \( k \), of the following form [8, 15]:

\[
f(k, i) = \beta_{0,k} + \beta_{1,k}x_{1,i} + \beta_{2,k}x_{2,i} + \cdots + \beta_{M,k}x_{M,i},
\]

where \( \beta_{m,k} \) is a regression coefficient associated with the \( m \)-th explanatory variable and the \( k \)-th effect. Explanatory variables and coefficients are organized into vectors of magnitude \( M + 1 \) so that the prediction function can be written in its most elaborate form [5]:

\[
f(k, i) = \beta_k \cdot x_i,
\]

where \( \beta_k \) is the set of coefficients related to the result \( k \), and \( x_i \) (vector line) is the set of explanatory variables related to the observation \( i \).

To arrive at the polynomial logit model, we use the logic that by running \( K - 1 \) independent regression accounting models, one result is selected as constant, and the other \( K - 1 \) results are regressed against the fixed result. If the result \( K \) (the last result) is chosen as the constant, then [5, 10]

\[
\ln \frac{\Pr(Y_i = 1)}{\Pr(Y_i = K)} = \beta_1 \cdot X_i,
\]

\[
\ln \frac{\Pr(Y_i = 2)}{\Pr(Y_i = K)} = \beta_2 \cdot X_i,
\]

\[
\ln \frac{\Pr(Y_i = K - 1)}{\Pr(Y_i = K)} = \beta_{K-1} \cdot X_i.
\]

For this reason, we defined different sets of regression constants, one for each possible result. Raising the power of \( e \)
on both sides of each equation and solving in terms of probabilities, we have [16, 17]

\[
\begin{align*}
\Pr(Y_i = 1) &= \Pr(Y_i = K) e^\beta_1 x_i, \\
\Pr(Y_i = 2) &= \Pr(Y_i = K) e^\beta_2 x_i, \\
\cdots \\
\Pr(Y_i = K - 1) &= \Pr(Y_i = K) e^\beta_{K-1} x_i.
\end{align*}
\]

(4)

Using the fact that the sum of all \( K \) probabilities must make 1, then

\[
\begin{align*}
\Pr(Y_i = K) &= 1 - \sum_{k=1}^{K-1} \Pr(Y_i = k) \\
&= 1 - \sum_{k=1}^{K-1} \Pr(Y_i = K) e^\beta_k x_i \\
&= \frac{1}{1 + \sum_{k=1}^{K-1} e^\beta_k x_i}
\end{align*}
\]

(5)

Respectively, we use the above to find the other possibilities [11, 18, 19].

\[
\begin{align*}
\Pr(Y_i = 1) &= \frac{e^\beta_1 x_i}{1 + \sum_{k=1}^{K-1} e^\beta_k x_i}, \\
\Pr(Y_i = 2) &= \frac{e^\beta_2 x_i}{1 + \sum_{k=1}^{K-1} e^\beta_k x_i}, \\
\cdots \\
\Pr(Y_i = K - 1) &= \frac{e^\beta_{K-1} x_i}{1 + \sum_{k=1}^{K-1} e^\beta_k x_i}.
\end{align*}
\]

(6)

The fact that we performed multiple regressions proves why we assumed the independence of irrelevant alternatives. Thus, the estimation of the desired distributed solution is feasible. An abstract illustration of the proposed architecture, based on how the schema can work in MDML, is shown in Figure 1.

By this logic, this model is a distributed machine learning technique as the proposed learning algorithm is implemented in multiple nodes to improve performance, increase accuracy, and distribute the input data of a learning model. This distributed nature of the proposed algorithm allows substantiated decisions to be made from large data sets [20, 21].

**Figure 1**: An abstract depiction of the proposed architecture.
and exploit them for financial gain \cite{23, 24}. Infrastructure to capture the accounts of unsuspecting users of streaming platforms. The most common tactic is to use botnets to steal credentials and generally launch ATA on music streaming applications. However, cybercriminals often utilize advanced techniques, even zero-day attacks, to create a new landscape of threats \cite{22}. By this logic, they can be extended to multiple models. The logarithm of the likelihood of selecting one class over another is independent of the presence or absence of other irrelevant choices \cite{29}. For example, if a bicycle is offered as an additional choice, the relative odds of choosing a car or bus to get to work do not vary. This enables the modeling of a set of K – 1 independent binary options as a collection of selected K alternatives. Each separate is chosen as a constant, and the other K – 1s are compared to it one by one. Although hypothesis IIA is a core hypothesis in rational choice theory, several psychological investigations reveal that people frequently breach this criterion when making decisions \cite{30, 31}. When the polynomial logit is used to model options, it might produce too much confusion between the meaningful choices and between different alternatives in some instances. This is critical if the study’s goal is to forecast how options would change if one of them disappeared. In comparable cases, other models, such as the nested logit or the polynomial probit, can be utilized because they enable the violation of the IIA.

As a log-linear model, binary accounting regression can be extended to multiple models. The logarithm of the separation function is obtained by modeling the logarithm of the relative odds of choosing a car or bus to get to work do not vary. This enables the modeling of a set of K – 1 independent binary options as a collection of selected K alternatives. Each separate is chosen as a constant, and the other K – 1s are compared to it one by one. Although hypothesis IIA is a core hypothesis in rational choice theory, several psychological investigations reveal that people frequently breach this criterion when making decisions \cite{30, 31}.

When the polynomial logit is used to model options, it might produce too much confusion between the meaningful choices and between different alternatives in some instances. This is critical if the study’s goal is to forecast how options would change if one of them disappeared. In comparable cases, other models, such as the nested logit or the polynomial probit, can be utilized because they enable the violation of the IIA.

As a log-linear model, binary accounting regression can be extended to multiple models. The logarithm of the separation function is obtained by modeling the logarithm of the relative odds of choosing a car or bus to get to work do not vary. This enables the modeling of a set of K – 1 independent binary options as a collection of selected K alternatives. Each separate is chosen as a constant, and the other K – 1s are compared to it one by one. Although hypothesis IIA is a core hypothesis in rational choice theory, several psychological investigations reveal that people frequently breach this criterion when making decisions \cite{30, 31}.

When the polynomial logit is used to model options, it might produce too much confusion between the meaningful choices and between different alternatives in some instances. This is critical if the study’s goal is to forecast how options would change if one of them disappeared. In comparable cases, other models, such as the nested logit or the polynomial probit, can be utilized because they enable the violation of the IIA.
As in the binary case, we need an additional term \(-\ln Z\) to ensure that all probabilities form a probability distribution so that they have a sum of 1.

\[
\sum_{k=1}^{K} \operatorname{Pr}(Y_i = k) = 1.
\] (8)

We need to add a condition to ensure normalization, in addition to the usual multiplication, because we have taken the logarithm of the probabilities [36]. By increasing the power of each member of the equations, we convert the prosthetic part into a multiplier, and the probability becomes the measure of Gibbs [37].

\[
\operatorname{Pr}(Y_i = 1) = \frac{1}{Z} e^{\beta_1 x_i},
\]

\[
\operatorname{Pr}(Y_i = 2) = \frac{1}{Z} e^{\beta_2 x_i},
\]

\[
\ldots
\]

\[
\operatorname{Pr}(Y_i = K) = \frac{1}{Z} e^{\beta_K x_i}.
\] (9)

The quantity \(Z\) is known as the distribution partition function. We can compute the value of this function by applying the preceding constraint, which demands that all probabilities have a sum of 1.

\[
1 = \sum_{k=1}^{K} \operatorname{Pr}(Y_i = k) = \sum_{k=1}^{K} \frac{1}{Z} e^{\beta_k x_i} = \frac{1}{Z} \sum_{k=1}^{K} e^{\beta_k x_i},
\]

\[
Z = \sum_{k=1}^{K} e^{\beta_k x_i}.
\] (10)

The coefficient is a constant because it is not a function of \(Y_i\), which is the variable thanks to which the probability distribution is defined. However, it is not variable about the explanatory variables or unknown \(\beta_k\) coefficients of regression, which should be determined through an optimization process. The final equations of probability are [38–40]

\[
\operatorname{Pr}(Y_i = 1) = \frac{e^{\beta_1 x_i}}{\sum_{k=1}^{K} e^{\beta_k x_i}},
\]

\[
\operatorname{Pr}(Y_i = 2) = \frac{e^{\beta_2 x_i}}{\sum_{k=1}^{K} e^{\beta_k x_i}},
\]

\[
\ldots
\]

\[
\operatorname{Pr}(Y_i = K) = \frac{e^{\beta_K x_i}}{\sum_{k=1}^{K} e^{\beta_k x_i}},
\]

\[
\operatorname{Pr}(Y_i = c) = \frac{e^{\beta_c x_i}}{\sum_{k=1}^{K} e^{\beta_k x_i}},
\]

\[
\operatorname{softmax}(k, x_1, \ldots, x_n) = \frac{e^{\beta_k x_i}}{\sum_{i=1}^{n} e^{\beta_i x_i}}
\]

is called a softmax function. The reason is that the exposure of the variables \((x_1, \ldots, x_n)\) magnifies the differences between them. As a result, softmax \((k, x_1, \ldots, x_n)\) [41, 42] will return a value close to 0 when \(x_k\) is much less than the maximum of all values and will return 1 when applied to the maximum value, unless very close to the next highest price.

This function may build a weighted average that acts like a smooth function (and can be easily separated) and values the index function.

\[
\begin{cases} 
1, & k = \arg \max_i (x_1, \ldots, x_n) \\
0, & \text{otherwise}
\end{cases}
\] (12)

Thus, we can write the probability equations as follows:

\[
\operatorname{Pr}(Y_i = c) = \operatorname{softmax}(c, \beta_1 \cdot X_1, \ldots, \beta_K \cdot X_i).
\] (13)

This function is the equivalent of the accounting function in binary accounting regression.

In general, there are only \(k - 1\) individually differentiable probabilities, and therefore, there are \(k - 1\) separately distinct coefficient vectors. One way to look at this is to see that if we add a constant vector to all factor vectors, then the equations are identical.

\[
\frac{e^{(\beta_c x_i)}}{\sum_{k=1}^{K} e^{(\beta_k x_i)}} = \frac{e^{(\beta_1 x_i) x_i}}{\sum_{k=1}^{K} e^{(\beta_k x_i) x_i}} = \frac{e^{(\beta_2 x_i) x_i}}{\sum_{k=1}^{K} e^{(\beta_k x_i) x_i}} = \frac{e^{(\beta_K x_i) x_i}}{\sum_{k=1}^{K} e^{(\beta_k x_i) x_i}}
\] (14)

As a result, it is common to set \(C = -\beta K\) (or someone other than the coefficient vector). Essentially, we set the variable so that one of the vectors becomes 0, and the remaining vectors are changed into the difference between these vectors and the vectors we chose. The coefficients are transformed mathematically as follows [43]:

\[
\beta_1 = \beta_1 - \beta_K,
\]

\[
\ldots
\]

\[
\beta_{i-1} = \beta_{i-1} - \beta_K,
\]

\[
\beta_K = 0.
\] (15)

This leads to the following equations [13, 35]:

\[
\operatorname{Pr}(Y_i = 1) = \frac{e^{\beta_1 x_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k x_i}},
\]

\[
\ldots
\]

\[
\operatorname{Pr}(Y_i = K-1) = \frac{e^{\beta_{K-1} x_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k x_i}},
\]

\[
\operatorname{Pr}(Y_i = K) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{\beta_k x_i}}
\] (16)

The distributed configuration, which is based on the configuration server and is adopted here, was acquired in 8 different contexts, and the results of the process are presented in Table 2.
The proposed method attempts to approach the amount of Pr(Y_i = 1) by performing the sampling for a few repetitions. Binary regression units are initialized to a sample of actual data and perform N iterations, taking some data due to the model contribution [10, 44, 45]. Essentially, the method undertakes to give lower importance to the real data and much higher energy in the cases where a marginal distribution comes from, thus helping the model to approach the actual data distribution and determine the nature of the ATA.

In conclusion, the proposed architecture can accurately learn N samples from N−1 models, while its learning speed can be even thousands of times faster than conventional methods. It is noteworthy that all βk coefficient vectors are uniquely recognizable. This has to do with the fact that all the odds must be added to 1, making one of them wholly decided while all the others are unknown. This can be translated, in combination with the very stable ability to predict, as the absence of contradictions between the models that act uniformly and respectively the existence of complete information and information about all the elements that make up the problem, so that the purpose of the decision comes from the choice of the optimal solution that maximizes the objective function and at the same time satisfies specific criteria.

5. Conclusions

In this work, we proposed an innovative MDML system, which significantly reduces the workload of servers in distributed environments and undertakes to perform machine learning tasks on large-scale data. Specifically, a distributed logit polynomial function model was proposed, which is used to model options in distributed binary regression accounting units. It is a highly efficient and low-demand distributed machine learning system tested to solve a highly demanding cybersecurity problem associated with distributed music repositories. Given the massive popularity of digital music repositories, targeted ATA attacks are carried out with malicious intent, such as stealing and removing personal information from user accounts.

As it turned out experimentally, the proposed system managed to categorize with great success the elements that lead to ATA attacks patterns. The high accuracy combined with the generalization of the results of the extensive tests indicates that this system is suitable for distributed environments and for solving highly complex problems. It is essential to say that the proposed standard had very stable predictability (0.9635 < F-score < 0.9818), which proves the great coherence of the models that act uniformly in recognition of standards and respectively the existence of complete information and information for all elements that make up the problem, even if they come from a distributed environment.

In the future, the proposed method can be extended by developing a mobile application that will work in an even more efficient and flexible federated style, allowing several users to contribute to the model training process. By merging unsupervised and supervised learning methodologies, the distributed training algorithm may be optimized and operated at an abstract level to handle more complex and multidimensional data in more extensive data sets. Furthermore, some bioinspired optimization approaches, such as the particle optimization methodology, can be utilized to develop solutions that maximize or reduce some study parameters, such as the cost function. Finally, it would be fascinating to use the polynomial logit function to create a distributed neural network in which kernel technology would be used to address issues with nonlinearity but time continuities.

Data Availability

The data used in this study are available from the author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References


