Credit Card Fraud Detection

Using the Machine Learning Classification Algorithms to detect Credit Card Fraudulent Activities

Introduction

- The Credit Card is a small plastic card issued to users as a System of Payment.

- Credit Card Security relies on the Physical Security of the plastic card as well as the privacy of the Credit Card Number.

- Globalization and increased use of the Internet for Online Shopping has resulted in a considerable proliferation of Credit Card Transactions throughout the world.

- Credit Card Fraud is a wide-ranging term for theft and fraud committed using a Credit Card as a fraudulent source of funds.

Logistic Regression Model

Logistic Regression Model

This slide is 100% editable. Adapt it to your needs and capture your audience's attention.

Text Here

This slide is 100% editable. Adapt it to your needs and capture your audience's attention.

Text Here

This slide is 100% editable. Adapt it to your needs and capture your audience's attention.

Text Here

This slide is 100% editable. Adapt it to your needs and capture your audience's attention.

A Classification of G-Invariant Shallow Neural Networks

Devanshu Agrawal & James Ostrowski

University of Tennessee, Knoxville

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Abstract

When trying to fit a deep neural network (DNN) to a G-invariant target function with G a group, it only makes sense to constrain the DNN to be G-invariant as well. However, there can be many different ways to do this, thus raising the problem of "G-invariant neural architecture design": What is the optimal G-invariant architecture for a given problem? Before we can consider the optimization problem itself, we must understand the search space, the architectures in it, and how they relate to one another. In this paper, we take a first step towards this goal; we prove a theorem that gives a classification of all G-invariant single-hidden-layer or "shallow" neural network (G-SNN) architectures with ReLU activation for any finite orthogonal group G, and we prove a second theorem that characterizes the inclusion maps or "network morphisms" between the architectures that can be leveraged during neural architecture search (NAS). The proof is based on a correspondence of every G-SNN to a signed permutation representation of G acting on the hidden neurons; the classification is equivalently given in terms of the first cohomology classes of G, thus admitting a topological interpretation. The G-SNN architectures corresponding to nontrivial cohomology classes have, to our knowledge, never been explicitly identified in the literature previously. Using a code implementation, we enumerate the G-SNN architectures for some example groups G and visualize their structure. Finally, we prove that architectures corresponding to inequivalent cohomology classes coincide in function space only when their weight matrices are zero, and we discuss the implications of this for NAS.

Introduction

- When fitting a deep neural network (DNN) to a target function that is known to be G-invariant with respect to a group G, it is desirable to enforce G-invariance on the DNN as prior knowledge.
- Numerous G-invariant DNN (G-DNN) architectures have been proposed over the years, including G-equivariant convolutional neural networks (G-CNNs) [1], G-equivariant graph neural networks [6], and a DNN stacked on a G-invariant sum-product layer [4].
- Despite a complete classification of G-CNNs [2, 5], it is only conjectured that every G-DNN is a G-CNN.
- It is unclear which G-DNN architecture should be used for a given problem, or even how to select the representations of G used in the layers of a G-CNN.

Problem statement

What is the "best" way to constrain the parameters of a DNN such that it is G-invariant for a given problem?

Longterm strategy

1. Classify all G-DNN architectures.
2. Characterize the inclusion maps or "network morphisms" [7] between the G-invariant architectures.
3. Perform *neural architecture search* (NAS) [3] to find the optimal G-DNN architecture.

Scope and contributions

- We consider only G-invariant single-hidden-layer or "shallow" neural networks (G-SNNs) with ReLU activation for finite orthogonal groups G acting on the input space.
- We classify all G-SNN architectures and the network morphisms between them but leave the implementation of NAS for future work.
- We discover *type 2* G-SNN architectures that have not appeared in the literature previously.

Signed permutation representations

- Let G be a finite orthogonal matrix group.
- A *signed permutation representation* (signed perm-rep) is a representation of G in terms of signed permutation matrices.
- For every $K \leq H \leq G$, $|H:K| \leq 2$, define the *paired conjugacy class*

$$(H,K)^G = \{(g^{-1}Hg, g^{-1}Kg) : g \in G\}.$$

Theorem 1. *The irreducible signed perm-reps ρ_{HK} of G, up to conjugation, are in one-one correspondence with the paired conjugacy classes $(H,K)^G$.*

Classification of G-SNNs

- Every G-SNN architecture is a sum of irreducible G-SNN architectures, and it is thus sufficient to classify the irreducibles.
- For every $H \leq G$, let P_H be the orthogonal projection operator whose range is the pointwise invariant space of G, and let $\mathrm{st}_G(P_H) = \{g \in G : gP_H = P_H\}$ be the stabilizer subgroup of P_H.

Theorem 2. *Every irreducible G-SNN architecture can be enumerated exactly once as follows: For every conjugacy pair $(H,K)^G$ such that $\mathrm{st}_G(P_K - (|H:K|-1)P_H) = K$, define the G-SNN architecture*

$$\begin{aligned} f_{HK}(x) &= a_*^\top \mathrm{ReLU}(W_* x + b_*) + c_*^\top x + d_*, \\ W_* &= \sum_{g_i H \in G/H} e_i (g_i w)^\top, \; w \in \mathrm{ran}(P_K - (|H:K|-1)P_H), \; \|w\| = 1 \\ a_* &= a\mathbf{1}, \; a \neq 0 \\ b_* &= (2 - |H:K|) b\mathbf{1}, \; b \in \mathbb{R} \\ c_* &= -\frac{1}{2}(|H:K|-1) W_*^\top a_* + c, \; c \in \mathrm{ran}(P_G). \end{aligned}$$

Proof sketch.

1. Classify the parameterization redundancies of an SNN.
2. Find a canonical parameterization of SNNs eliminating these redundancies.
3. The canonical parameters of a G-SNN are invariant under the action of G and thus satisfy a set of constraints, given in terms of a signed perm-rep.
4. Classify all irreducible signed perm-reps of G, and enumerate these constraint sets.

□

- The architecture f_{HK} is said to be of *type* $|H:K|$, and to our knowledge, the type 2 architectures have never appeared in the literature previously.
- The hidden neurons of f_{HK} transform according to the signed perm-rep ρ_{HK} corresponding to $(H,K)^G$ under the action of G.
- The stabilizer subgroup condition helps to exclude G-SNN architectures with redundant weight vectors.

Network morphisms

- We say there is a network morphism from G-SNN architecture f_1 to architecture f_2, and write $f_1 \mapsto f_2$, if for every instantiation $f_1(\cdot;\theta)$ of f_1, there exists a sequence of instantiations $\{f_2(\cdot;\theta_n)\}_{n\geq 1}$ of f_2 such that $f_2(\cdot;\theta_n) \to f_1(\cdot;\theta)$ in the topology of uniform convergence on compact sets.
- These network morphisms can be leveraged during NAS to map one architecture into another, e.g., if the former is underfitting.
- The definition of "network morphism" is functional-analytic, but the following theorem gives a group-theoretic characterization.

Theorem 3. *Let $(H_i,K_i)^G$, $i \in \{1,2\}$, be two paired conjugacy classes. Then $f_1 \mapsto f_2$ iff there exists $(H,K) \in (H_2,K_2)^G$ such that $H \leq H_1$, $K \leq K_1$, and $H \cap K_1 = K$.*

- The proof relies on the Arzelà-Ascoli Theorem and a non-canonical parameterization of G-SNNs.
- As a corollary, G-SNN architecture space is connected.

Topological tunneling

- The following proposition states that G-SNN architectures corresponding to distinct "cohomology classes" of a signed perm-rep are in a sense orthogonal.

Proposition 4. *Let w_1 and w_2 be the top canonical weight vectors of two irreducible G-SNN architectures f_{HK_1} and f_{HK_2} where $K_1 \neq K_2$. Then $w_1^\top w_2 = 0$.*

- We call the resulting phenomenon *topological confinement*, and it implies that two architectures with the same unsigned permutation pattern (subgroup H) but different sign-flip patterns (subgroup K) cannot approximate one another.
- We propose to use *topological tunneling* during NAS, where the choice of cohomology class (subgroup K) can be transformed to provide shortcuts in architecture space, thus potentially speeding up the search.

Example: Cyclic rotation group

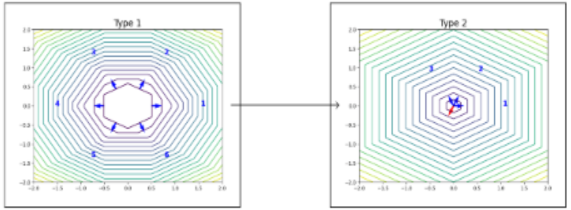


Figure 1. Irreducible G-SNN architecture space for the group G of 60° rotations on the input space $\mathbb{R}^2$. The contour plot of each architecture as a function $\mathbb{R}^2 \to \mathbb{R}$ is displayed. The blue vectors are the weight vectors of W_*, and their offsets from the origin in the type 1 architecture indicate the bias b_*. The red vector in the type 2 architecture is the parameter c_* of the G-SNN. The black arc between the architectures represents a network morphism.

Example: Cyclic permutation group

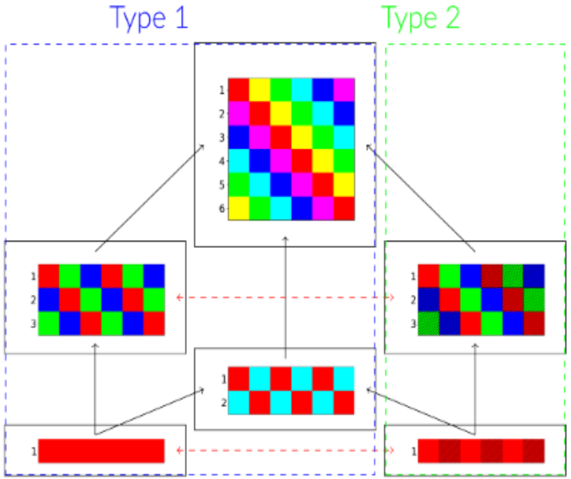


Figure 2. Irreducible G-SNN architecture space for the group G of all cyclic permutations of the dimensions of the input space $\mathbb{R}^6$. The weight-sharing pattern of W_* for each architecture is displayed, where the number of rows is the number of hidden neurons in the architecture. Weights of the same color and texture (solid vs. hatched) are constrained to be equal; weights of the same color but different texture are constrained to be opposites (colors should not be compared across different architectures). Every directed path of black arcs represents a network morphism, and the red arcs represent topological tunneling.

Architecture count for small groups

Table 1. Ratio of the number of irreducible G-SNN architectures to the number of irreducible signed perm-reps of each type (1 vs. 2) for every group G, $|G| \leq 8$, up to isomorphism. We considered only one particular representation for each group.

G	Type 1	Type 2
$\{e\}$	1/1	0/0
C_2	2/2	1/1
C_3	2/2	0/0
C_4	3/3	2/2
C_5	2/2	0/0
C_6	4/4	2/2
C_7	2/2	0/0
C_8	4/4	3/3
C_2^2	4/5	3/6
C_2^3	8/16	7/35
$C_2 \times C_4$	6/8	5/11
D_3	3/4	1/2
D_4	5/8	7/13
Q_8	6/6	7/7

Conclusion and future directions

Conclusion

- Theorems 2-3 together give a complete description of the structure of G-SNN architecture space and thus establish the theoretical foundation for *G-invariant NAS (G-NAS)*.
- Our theory not only organizes G-SNN architectures already known into a space but also reveals the existence of the type 2 architectures not known previously.

Future directions

- We will extend our theory to deep architectures.
- We will implement and investigate G-NAS in practice.

Key references

[1] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[2] Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. *Advances in neural information processing systems*, 32, 2019.

[3] Thomas Elsken, Jan Hendrik Metzen, Frank Hutter, et al. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20(55):1–21, 2019.

[4] Piotr Kicki, Mete Ozay, and Piotr Skrzypczyński. A computationally efficient neural network invariant to the action of symmetry subgroups. *arXiv preprint arXiv:2002.07528*, 2020.

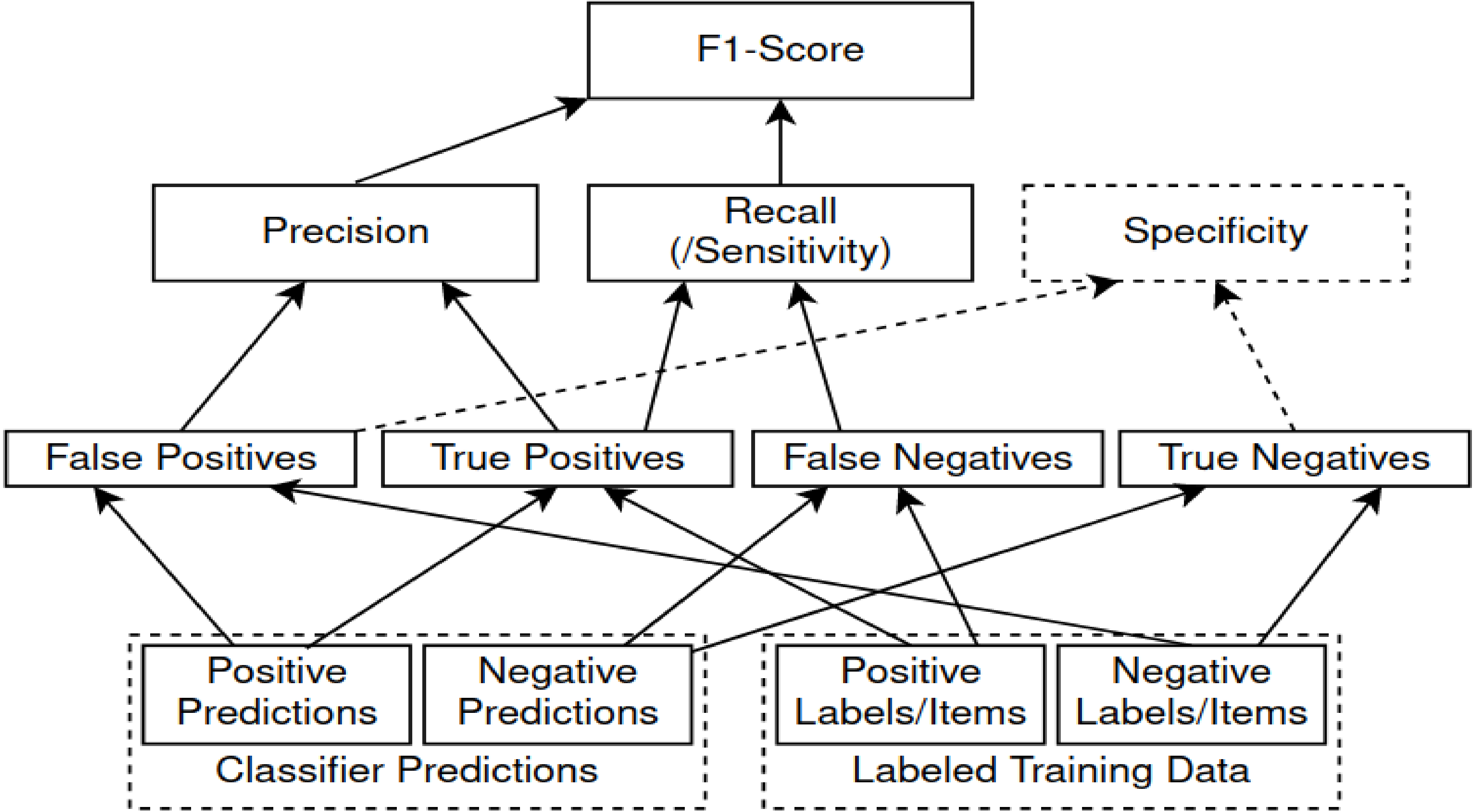
[5] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2747–2755. PMLR, 2018.

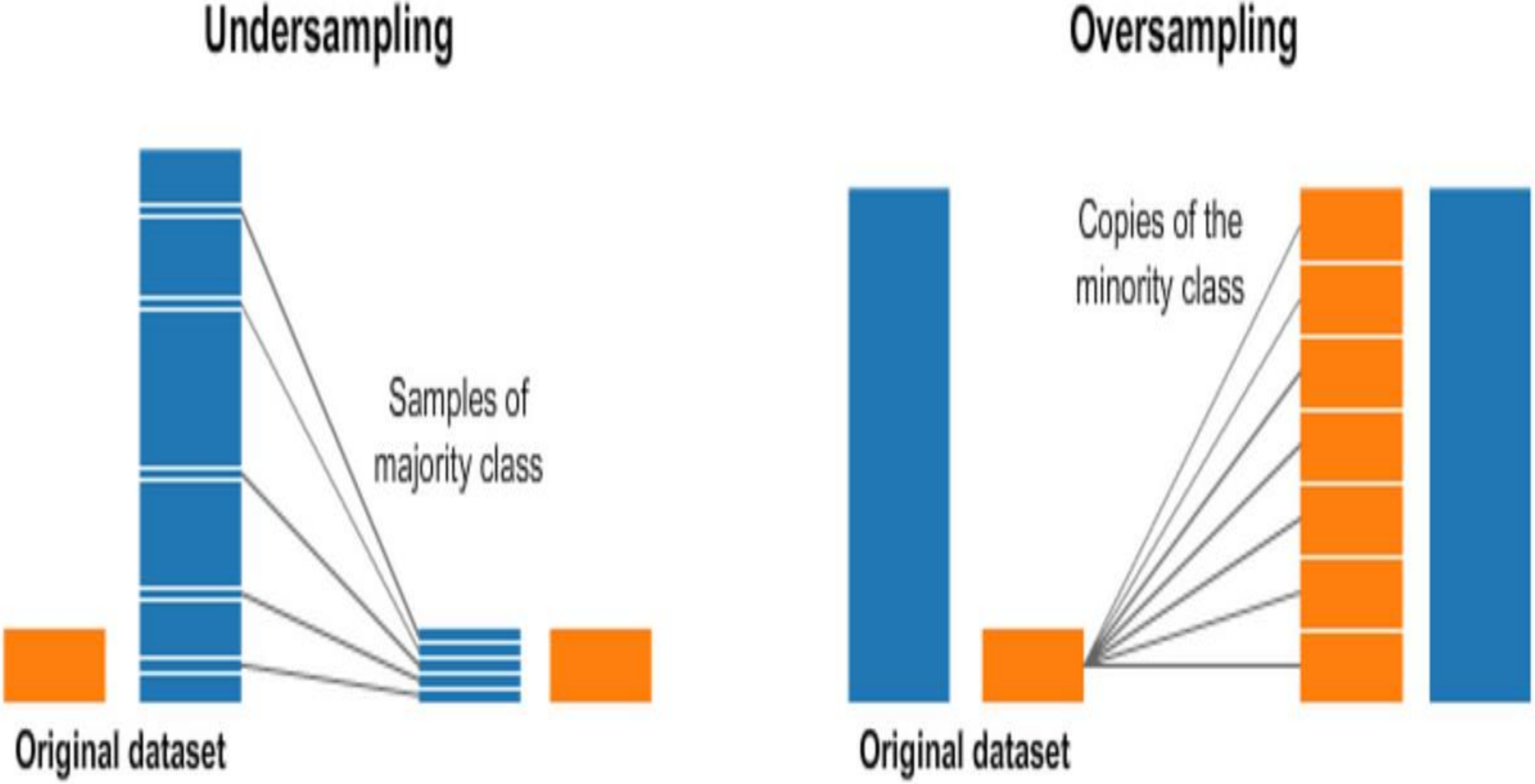
[6] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. volume 7, 2019.

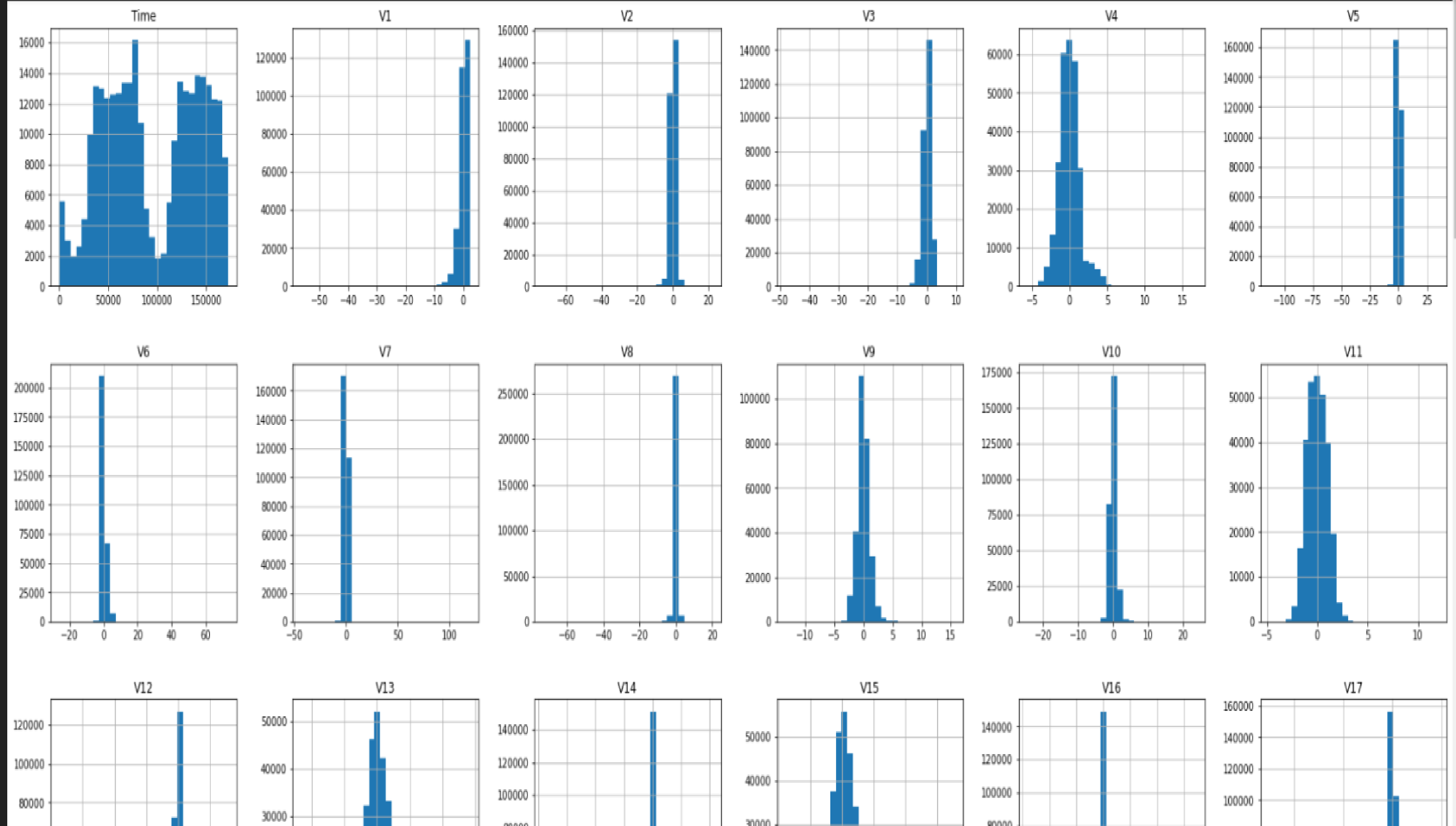
[7] Tao Wei, Changhu Wang, Yong Rui, and Chang Wen Chen. Network morphism. In *International Conference on Machine Learning*, pages 564–572. PMLR, 2016.

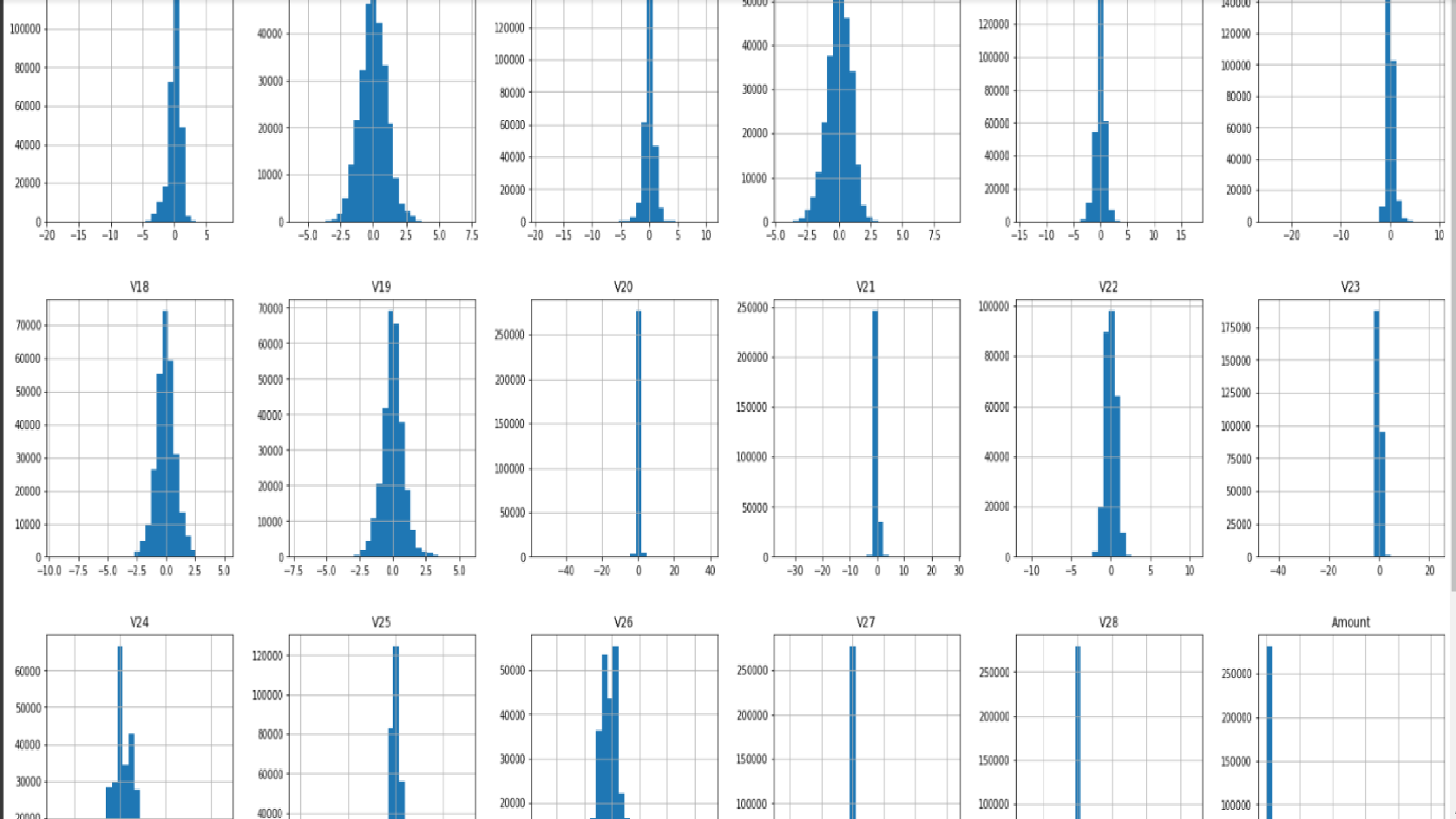
Acknowledgments

D.A. and J.O. were supported by DOE grant DE-SC0018175. D.A. was additionally supported by NSF award No. 2202990.


F1-Score
Precision
Recall
(/Sensitivity)
Specificity
False Positives
True Positives
False Negatives
True Negatives
Positive
Predictions
Negative
Predictions
Positive
Labels/Items
Negative
Labels/Items
Classifier Predictions
Labeled Training Data


Undersampling
Samples of majority class
Original dataset
Oversampling
Copies of the minority class
Original dataset


Time
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
V17



V18
V19
V20
V21
V22
V23
V24
V25
V26
V27
V28
Amount
−20 −15 −10 −5 0 5
−5.0 −2.5 0.0 2.5 5.0 7.5
−20 −15 −10 −5 0 5 10
−5.0 −2.5 0.0 2.5 5.0 7.5
−15 −10 −5 0 5 10 15
−20 −10 0 10
−10.0 −7.5 −5.0 −2.5 0.0 2.5 5.0
−7.5 −5.0 −2.5 0.0 2.5 5.0
−40 −20 0 20 40
−30 −20 −10 0 10 20 30
−10 −5 0 5 10
−40 −20 0 20

Thank you!