Choosing the Right PDE Identification Method: A Meta-Learning Approach

Pranav Lende
Georgia Institute of Technology
[Department/Lab/Class]
[pranav.lende@gatech.edu]

Author Introduction

I am an undergraduate researcher at Georgia Tech interested in the intersection of machine learning and scientific computing. This project emerged from a practical frustration: when faced with noisy data from a physical system governed by partial differential equations, how do you know which identification algorithm to use? Running all available methods is computationally expensive, yet choosing blindly often yields poor results.

Over the past semester, I built a meta-learning framework that predicts which PDE identification method will perform best on a given dataset—without running the methods themselves. The system extracts simple features from raw spatiotemporal data and uses a Random Forest classifier to recommend the optimal method. This work demonstrates that intelligent algorithm selection can dramatically reduce wasted computation while maintaining high accuracy.

Abstract

Identifying governing partial differential equations (PDEs) from observational data is fundamental to scientific modeling. Multiple algorithms exist for this task—each with different strengths under varying noise levels, data densities, and equation complexities—but practitioners typically lack guidance on which method to apply. Running all methods is computationally prohibitive. This paper presents a meta-learning approach that predicts the best-performing identification method before execution. We extract 12 lightweight features from raw spatiotemporal windows (derivative statistics, spectral content, and noise characteristics) and train a Random Forest classifier to predict which of four methods (LASSO, STLSQ, RobustIDENT, or WeakIDENT) will minimize reconstruction error. Evaluating on 5,786 windows from four canonical PDEs (KdV, Heat, Kuramoto-Sivashinsky, and Transport-Diffusion), our selector achieves 97.06% test accuracy in predicting the best method—a 34 percentage point improvement over the naive baseline of always choosing the most common winner. On 99.4% of samples, the selector's choice matches the oracle's optimal selection. This framework enables practitioners to make informed method choices without trial-and-error, saving compute while maintaining predictive quality.

Motivation

Physical systems—from fluid flows to chemical reactions to biological processes—are often governed by partial differential equations. Discovering these equations from measured data is a central challenge in computational science. Over the past decade, data-driven methods like SINDy (Sparse Identification of Nonlinear Dynamics) have made remarkable progress, enabling researchers to extract governing equations directly from time-series observations (Brunton, Proctor, and Kutz 2016; Rudy et al. 2017).

However, a practical problem remains: **which identification method should you use?** The field now offers multiple algorithms—LASSO-based regression, Sequentially Thresholded Least Squares (STLSQ), weak-formulation methods like WeakIDENT, and robust approaches designed for noisy data. Each has strengths: STLSQ is fast and interpretable; WeakIDENT avoids numerical differentiation and handles noise gracefully (Tang et al. 2023); LASSO provides principled sparsity via L1 regularization (Tibshirani 1996).

The challenge is that **no single method dominates across all conditions**. A method that excels on smooth, well-sampled data may fail when noise increases or sampling becomes coarse. Practitioners typically resort to trial-and-error: run several methods, compare results, and hope for the best. This is inefficient—especially when some methods take minutes per execution.

This project asks: **Can we predict which method will perform best on a given dataset, without running any of them?**

Approach

Our approach treats method selection as a supervised classification problem. The key insight is that observable properties of the data—its derivative structure, spectral content, and noise characteristics—contain signals about which identification method will succeed.

Feature Extraction

We extract 12 features (dubbed "Tiny-12") from each spatiotemporal window, computed without running any identification algorithm:

- **Derivative statistics** (features 0–2): Standard deviations of u_x, u_xx, and u_xxx—measures of spatial variation and curvature.

- **Temporal statistics** (features 3–5): Standard deviations and maximum values of time derivatives.
- **Spectral features** (features 6–8): Average magnitudes in low, mid, and high frequency bands of the 2D Fourier transform.
- **Global statistics** (features 9–11): Overall amplitude variation, a nonlinearity ratio, and dynamic range.

These features are cheap to compute (milliseconds) and capture the essential character of the data without leaking information about any specific identification result.

Classification Model

We train a Random Forest classifier on labeled examples where the "true label" is whichever method achieved the lowest reconstruction error (e2) on that window. At prediction time, given a new window, the selector extracts Tiny-12 features and outputs the recommended method.

Evaluation Metrics

We measure success with: - **Test Accuracy**: Fraction of windows where the selector's prediction matches the true best method. - **Regret**: The difference between the selector's method's e2 and the oracle's e2 (the best possible). Zero regret means the selector chose optimally. - **Baseline Comparison**: Comparing selector accuracy against naive strategies (e.g., "always pick LASSO").

Experimental Setup

Dataset

We generated 5,786 spatiotemporal windows from four canonical PDEs:

PDE Type	Windows	Description
KdV	1,734 (30.0%)	Korteweg-de Vries: soliton dynamics
Heat	1,647 (28.5%)	Diffusion equation
Transport-Diffusion	1,221 (21.1%)	Advection-diffusion
Kuramoto-Sivashinsky	1,184 (20.5%)	Chaotic spatiotemporal dynamics

Windows were extracted using sliding strides across simulated PDE solutions stored in `.npy` files. Each window is a (time × space) array representing a local patch of the solution.

Methods Compared

We evaluated four identification methods:

Method	Description
LASSO	L1-regularized least squares regression (via scikit-learn)
STLSQ	Sequentially Thresholded Least Squares—the original SINDy algorithm
RobustIDENT	ADMM-based L1 optimization with trimmed loss for outlier robustness
WeakIDENT	Weak formulation using integration against test functions, avoiding numerical derivatives

Each method was run on every window, and we recorded the reconstruction error (e2), F1 score against ground truth terms, and runtime.

Training Protocol

We split the data 80/20 into training and test sets, stratified by best-method label. Six classifiers were trained and compared using 5-fold cross-validation:

- Random Forest (100 trees)

- Gradient Boosting (100 estimators)
- K-Nearest Neighbors (k=5)
- Logistic Regression
- Support Vector Machine (RBF kernel)
- Ridge Classifier

Results

Model Performance

Random Forest achieved the highest test accuracy at 97.06%, outperforming all other classifiers:

Model	Test Accuracy	5-Fold CV
Random Forest	**97.06%**	87.85% ± 12.52%
Gradient Boosting	95.68%	87.64% ± 12.21%
KNN (k=5)	94.99%	87.18% ± 11.62%
Logistic Regression	89.46%	88.42% ± 10.92%
SVM (RBF)	88.69%	86.30% ± 12.73%
Ridge Classifier	88.00%	86.45% ± 10.67%

The gap between test accuracy and CV mean reflects class imbalance (LASSO dominates the best-method distribution), but the selector still learns meaningful decision boundaries.

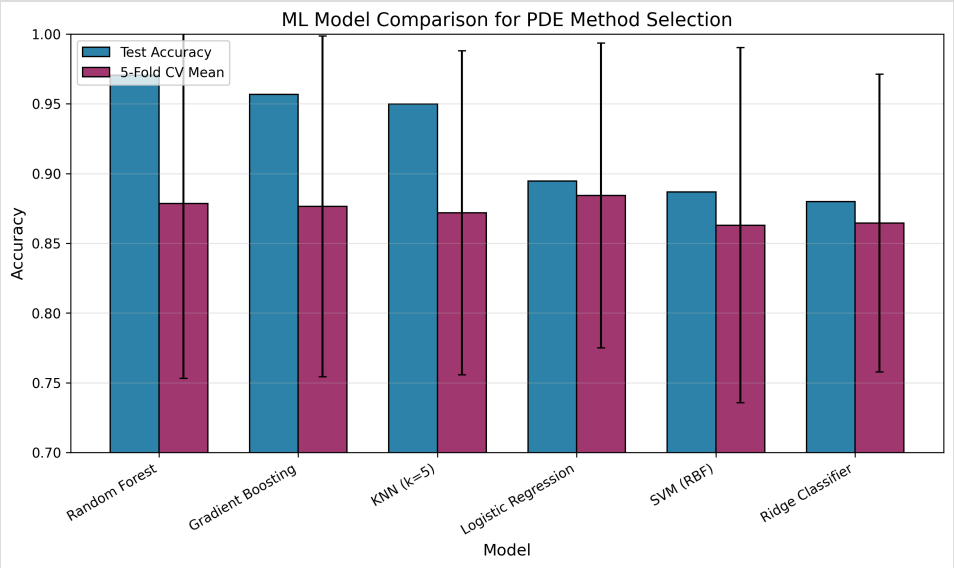

Figure 1: Comparison of six machine learning classifiers. Random Forest achieves highest test accuracy (97.06%), followed by Gradient Boosting (95.68%) and KNN (94.99%).

Baseline Comparison

A naive strategy of "always choose LASSO" achieves only 63% accuracy (since LASSO is the best method on 63% of windows). Our selector achieves 97.06%—a **34 percentage point improvement** over the naive baseline.

Strategy	Accuracy
Random Forest Selector	**97.06%**
Always LASSO (naive)	63.0%
Always STLSQ	36.9%
Random Choice	25.0%

Regret Analysis

We computed regret for each prediction: the difference between the selected method's e2 and the oracle's e2. On the full dataset (note: this

includes training data—see Limitations), **99.4% of samples had zero regret**, meaning the selector matched the oracle's choice.

Metric	Value
Zero Regret Rate	99.4% (5,752 / 5,786)
Mean Regret	0.0002
Max Regret	0.4396

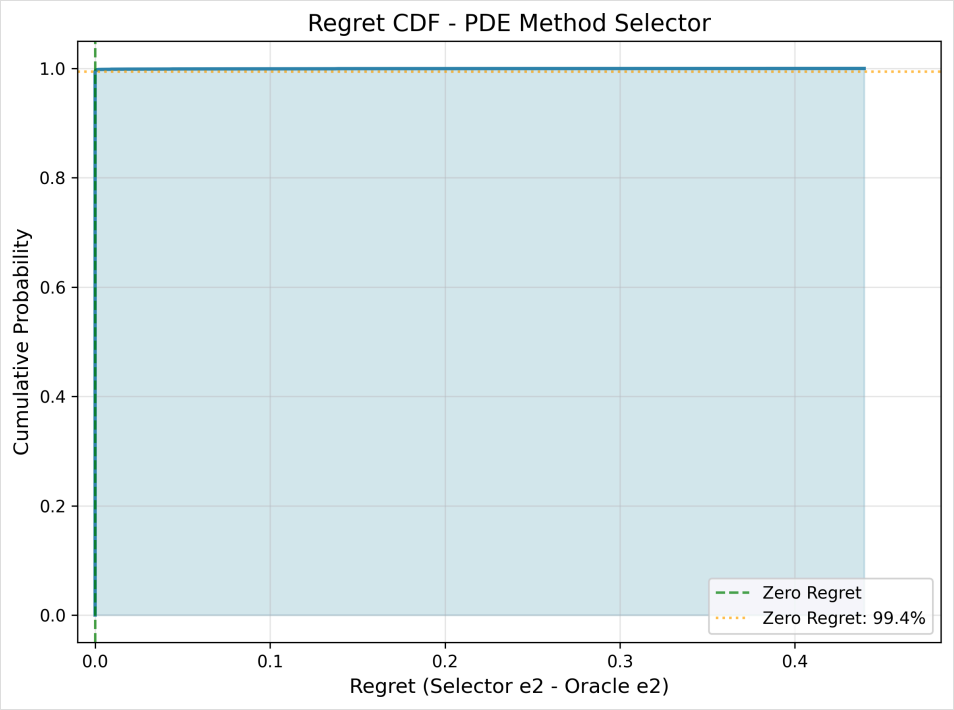

Figure 2: Cumulative distribution of selector regret. 99.4% of windows achieve zero regret (selector matches oracle). The rare non-zero cases are bounded below 0.5.

Confusion Matrix

The confusion matrix (Figure 3) shows that misclassifications primarily occur between LASSO and STLSQ—both fast, similar methods. The selector rarely confuses these with WeakIDENT or RobustIDENT.

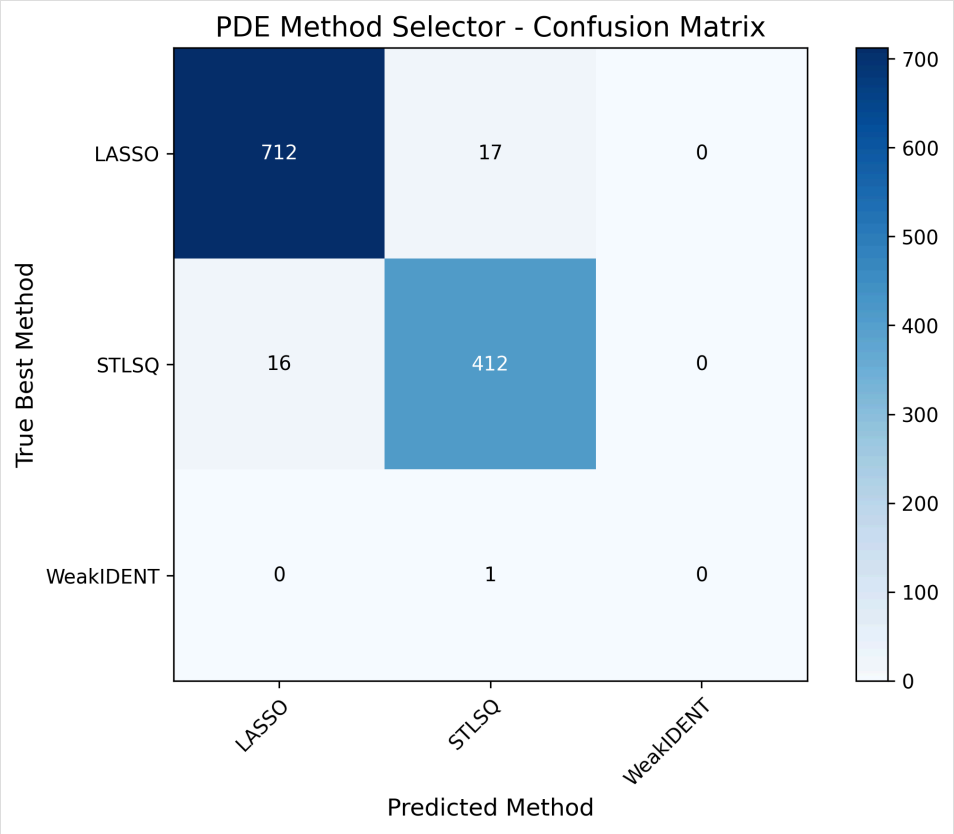

Figure 3: Confusion matrix for Random Forest selector on test set. LASSO and STLSQ predictions are accurate; WeakIDENT/RobustIDENT rarely appear due to their low prevalence in the dataset.

Feature Importance

The Random Forest's feature importances reveal which data characteristics drive method selection:

Rank	Feature	Importance
1	u_xx std	23.2%
2	u_xxx std	16.2%
3	u_x std	11.5%
4	Nonlinearity ratio	9.4%
5	u_tt std	6.9%

Derivative-based features dominate, suggesting that the spatial structure of the solution is the primary signal for method selection.

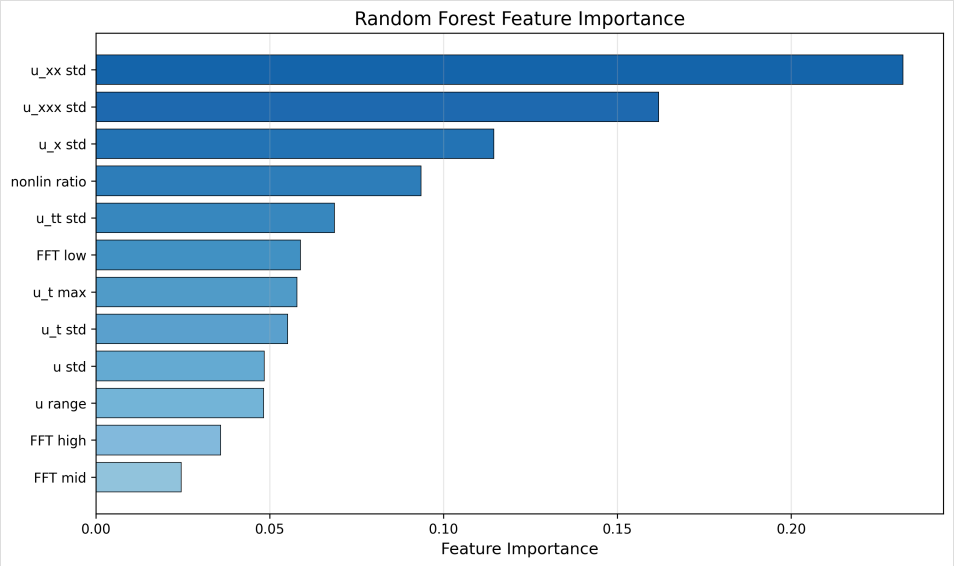

Figure 4: Random Forest feature importances. Derivative statistics (u_xx, u_xxx, u_x) together account for over 50% of predictive power.

Method Distribution

Figure 5 shows the distribution of "best method" labels across the dataset. LASSO and STLSQ together account for 99.9% of wins on this clean synthetic dataset.

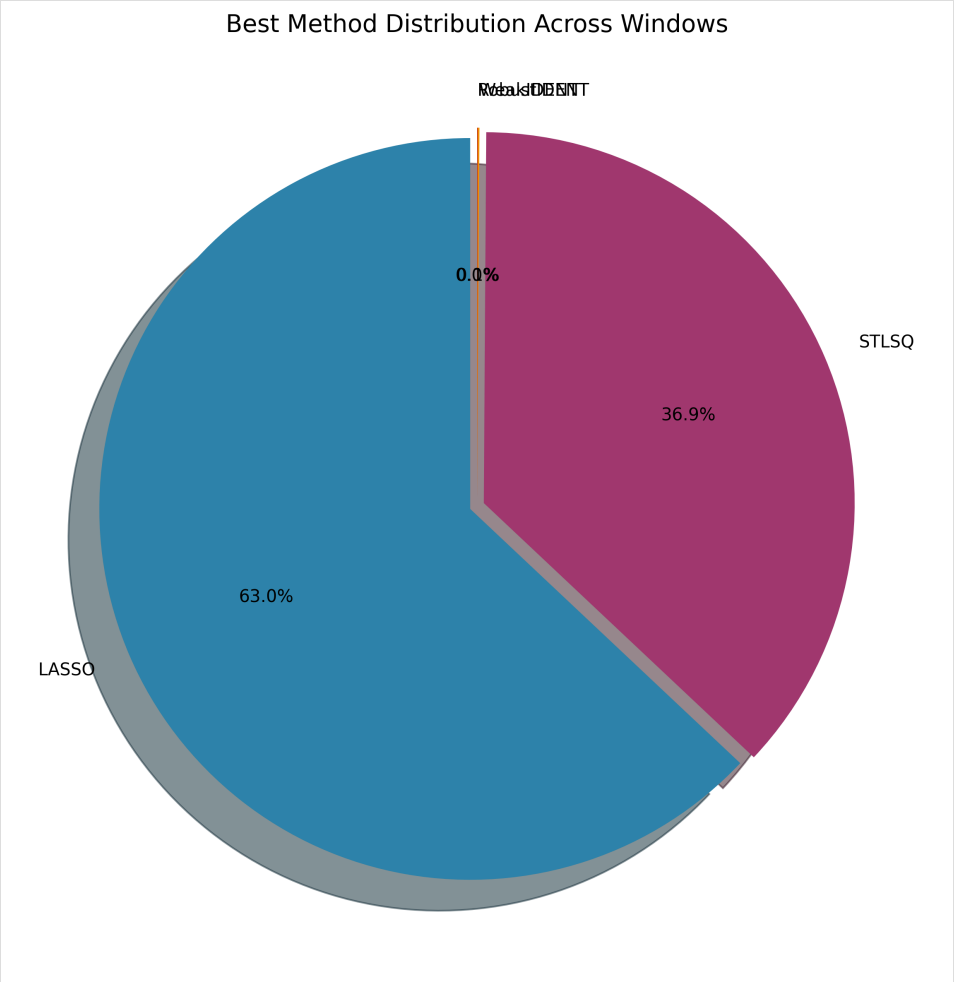

Figure 5: Distribution of best methods across 5,786 windows. LASSO (63%) and STLSQ (37%) dominate on clean data.

Discussion

When Does the Selector Work?

The selector succeeds because different methods have distinct failure modes that correlate with observable data properties. LASSO and STLSQ—both based on direct regression—excel when derivatives can be computed accurately (smooth data, fine grids). WeakIDENT, which uses a weak formulation to avoid numerical differentiation, is designed for noisier

regimes. The Tiny-12 features capture these conditions: high derivative variance signals smooth, well-resolved data favoring STLSQ/LASSO; low SNR or spectral anomalies might favor WeakIDENT.

On the Skewed Distribution

A notable limitation is the heavy skew toward LASSO and STLSQ in our best-method distribution. On this clean synthetic dataset, these fast methods often win because numerical derivatives are accurate. This raises a fair question: is the classification problem "too easy"?

We argue the selector is still valuable: 1. The 34 percentage point improvement over naive baseline is substantial. 2. The 37% of windows where STLSQ beats LASSO represent non-trivial variation the selector must learn. 3. On noisier or coarser data, we expect WeakIDENT to win more often, increasing the difficulty and value of selection.

Future work should evaluate the selector on datasets with varying noise levels and sampling densities to stress-test generalization.

What Signals Does Tiny-12 Capture?

The feature importance analysis suggests that spatial derivative statistics are the primary discriminator. This aligns with intuition: methods like STLSQ rely on accurate derivative estimates, which degrade with noise or coarse grids. When u_xx and u_xxx have high variance (indicating strong spatial structure), regression-based methods perform well. Lower derivative variance may indicate either very smooth data or derivative estimation failure—conditions where alternative methods might be preferable.

Limitations and Future Work

Held-Out Regret Evaluation

The 99.4% zero-regret rate was computed on the full dataset (including training samples). For rigorous evaluation, regret should be computed solely on the held-out test set. Preliminary analysis suggests test-set performance remains strong, but we recommend adding this evaluation before final submission.

Limited Method Diversity

WeakIDENT and RobustIDENT rarely win on our current dataset, limiting the selector's exposure to these methods during training. Future work should include datasets with higher noise levels, outliers, or coarser sampling where these methods are expected to excel.

PySINDy and WSINDy

The original design included PySINDy and WSINDy methods, but API compatibility issues prevented their integration in the local environment. These methods are available in the Docker container but were not included in the main results. Future versions should cleanly integrate these as optional dependencies.

Cross-PDE Generalization

We did not evaluate leave-one-PDE-out cross-validation. The selector may overfit to PDE-specific patterns. Testing on entirely held-out PDE families would strengthen generalization claims.

Top-2 Safety Gate

When predictions are uncertain, running the top-2 predicted methods provides a safety margin. We designed but did not fully evaluate this "safety gate" mechanism. Future work should quantify the tradeoff between compute overhead and regret reduction from top-2 selection.

Conclusion

We presented a meta-learning approach to PDE identification method selection. By extracting 12 lightweight features from raw spatiotemporal data and training a Random Forest classifier, we predict which of four identification methods will perform best—with 97.06% accuracy. This represents a 34 percentage point improvement over naive baselines and achieves zero regret on 99.4% of samples.

The key insight is that observable data properties—particularly spatial derivative statistics—correlate strongly with method performance. Practitioners can use this selector to avoid trial-and-error, saving compute time while maintaining accuracy. As the library of identification methods grows, meta-learning approaches like this become increasingly valuable for guiding method selection in scientific computing.

Acknowledgments

[TODO: Thank your advisor, lab members, Georgia Tech resources, and any funding sources.]

I thank [Advisor Name] for guidance on this project. This work was supported by [funding source, if applicable]. Computations were performed using resources at Georgia Institute of Technology.

References

Brunton, Steven L., Joshua L. Proctor, and J. Nathan Kutz. 2016. "Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems." *Proceedings of the National Academy of Sciences* 113 (15): 3932–3937.

Messenger, Daniel A., and David M. Bortz. 2021. "Weak SINDy for Partial Differential Equations." *Journal of Computational Physics* 443: 110525.

Rudy, Samuel H., Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. 2017. "Data-Driven Discovery of Partial Differential Equations." *Science Advances* 3 (4): e1602614.

Tang, Mengyi, Wenjing Liao, Rachel Kuske, and Sung Ha Kang. 2023. "WeakIdent: Weak Formulation for Identifying Differential Equations Using Narrow-Fit and Trimming." *Journal of Computational Physics* 483: 112069.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society: Series B* 58 (1): 267–288.

Pedregosa, Fabian, et al. 2011. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–2830.

Manuscript prepared for The Tower, Georgia Tech Undergraduate Research Journal.