Intrusion detection using logistic regression: A comparison of regularization methods and PCA-logistic regression method

Karan Goel
School of Computing and Information Technology
University of Wollongong
CSCI933, SN:7836685, kg956@uowmail.edu.au

April 19, 2024

Abstract

Abstract

This study explores the application of logistic regression to intrusion detection, framing it as a binary classification problem within the context of Internet of Things (IoT) environments. We investigate various regularization techniques—lasso, ridge, and elastic-net—to assess their effectiveness in model performance enhancement, complemented by a dimensionality reduction approach using Principal Component Analysis (PCA). Utilizing a comprehensive real-time IoT dataset, the models demonstrate high accuracy, with the best performing model, which integrates ridge regularization and the Newton-CG solver, achieving an accuracy of 98.6%. Additionally, models employing PCA to reduce feature dimensionality also showed promising results with accuracy of 98.8%, maintaining high accuracy while reducing computational complexity.

1 Introduction

In today's digitally interconnected landscape, safeguarding computer networks against malicious activities is an imperative task. The proliferation of network devices across various sectors, including healthcare, manufacturing, and agriculture, has led to an unprecedented surge in connectivity and data exchange. However, this interconnectedness exposes networks to potential exploitation by cybercriminals, posing significant threats to data integrity and network security.

As organizations increasingly rely on digital infrastructure to streamline operations, the need for robust intrusion detection mechanisms has become paramount. In this context, integrating machine learning techniques has garnered substantial attention for their potential to enhance the efficacy of intrusion detection systems. Our study delves into machine learning-driven intrusion detection, focusing on the application of logistic regression models.

Our primary objective is to evaluate the efficiency and effectiveness of logistic regression in accurately classifying network traffic as normal or indicative of a potential attack. Additionally, we will assess the trade-offs between model complexity, interpretability, and predictive performance to determine the optimal configuration for real-world applications.

This study evaluates several logistic regression methodologies for intrusion detection, including: **1) No regularization**, using a basic model setup; **2) L2 regularization (ridge)**, which mitigates overfitting by adding a penalty term that encourages smaller parameter values, enhancing model generalization; **3) L1 regularization (lasso)**, which promotes model simplicity and feature selection by penalizing the absolute values of regression coefficients; **4) L1-L2 regularization (elastic-net)**, which combines L1 and L2 to balance sparsity and parameter shrinkage, useful for correlated features; and **5) PCA for dimensionality reduction**, which projects features onto a lower-dimensional space to enhance computational efficiency and address the curse of dimensionality.

Existing Works

Numerous studies have endeavored to enhance intrusion detection systems through a variety of machine learning techniques, among them logistic regression. For example, Wang et al. Wang (2005) introduced an innovative ensemble approach utilizing a multinomial logistic regression model tailored to the most prevalent attack types. Similarly, Gu et al. Gu (2020) devised a method that combines

partial least squares regression and logistic regression with feature augmentation to achieve comparable results.

Shyla, Bhatnagar, Bali, and Bali (2022) proposed adaptive moment estimation-stochastic gradient descent with Ridge classifier, Logistic Regression, and an ensemble method in terms of both time complexity and accuracy.

Although our study does not employ any of the aforementioned methods in the realm of intrusion detection, it's important to acknowledge the diverse approaches that have been explored in the field.

2 Theory and properties of regression

Regression is a statistical method by which one variable is explained or understood on the basis of one or more other variables. The variable that is being explained is called the dependent, or response, variable; the other variables used to explain or predict the response are called independent variables. Hilbe (2009)

A regression problem involves finding a hypothesis h from a set of functions mapping X to Y, given a labeled sample S where inputs are drawn from an unknown distribution D. The objective is to minimize the expected loss or generalization error $R(h)$ with respect to the target function f. The loss function L measures the error between predicted and actual labels, commonly using squared error for $L(y, \hat{y}) = ||y - \hat{y}||^2$. The generalization error $R(h)$ is defined as the expected value of the loss function over the distribution D. The empirical loss $\hat{R}(h)$ is computed as the average loss over the labeled sample S. These are mathematically expressed as:

$$R(h) = \mathbb{E}_{x \sim D}[L(h(x), f(x))] \tag{1}$$

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} L(h(x_i), y_i) \tag{2}$$

Here, x_i and y_i represent the input-output pairs in the labeled sample, and $h(x_i)$ is the predicted output by the hypothesis h. Alan J. (2008)

2.1 L1, L2 and Elastic-net penalties

Regularization techniques such as L1, L2, and Elastic Net are widely used in machine learning and statistics to prevent overfitting and improve the generalization of predictive models.

L1 Regularization (Lasso Regression): L1 regularization adds a penalty term to the cost function equal to the absolute value of the magnitude of coefficients.

$$J(\mathbf{w}) = \text{Cost}(\mathbf{w}) + \lambda \sum_{i=1}^{n} |\mathbf{w}_i|$$

where $\text{Cost}(\mathbf{w})$ is the original cost function, λ is the regularization parameter, and $|\mathbf{w}_i|$ represents the absolute value of the i-th coefficient.

L1 regularization encourages sparse solutions by driving some coefficients to exactly zero. This leads to feature selection, making it effective when dealing with datasets containing many irrelevant features. Alan J. (2008)

L2 Regularization (Ridge Regression): L2 regularization adds a penalty term to the cost function equal to the squared magnitude of coefficients.

$$J(\mathbf{w}) = \text{Cost}(\mathbf{w}) + \lambda \sum_{i=1}^{n} \mathbf{w}_i^2$$

where $\text{Cost}(\mathbf{w})$ is the original cost function, λ is the regularization parameter, and $\mathbf{w}_i^2$ represents the squared value of the i-th coefficient.

L2 regularization penalizes large coefficients and shrinks them towards zero, but it rarely leads to zero coefficients. It helps in reducing the impact of irrelevant features and can improve the numerical stability of the solution. Alan J. (2008)

Elastic Net Regularization: Elastic Net combines L1 and L2 penalties by adding both penalties to the cost function.

$$J(\mathbf{w}) = \text{Cost}(\mathbf{w}) + \lambda_1 \sum_{i=1}^{n} |\mathbf{w}_i| + \lambda_2 \sum_{i=1}^{n} \mathbf{w}_i^2$$

where $\text{Cost}(\mathbf{w})$ is the original cost function, λ_1 and λ_2 are the regularization parameters for L1 and L2 penalties respectively.

Elastic Net combines the benefits of both L1 and L2 regularization. It encourages sparsity like L1 regularization while also handling multicollinearity among features like L2 regularization. The solution weight vector is modified by both penalties, resulting in a balance between feature selection and coefficient shrinkage. Alan J. (2008)

2.2 Logistic regression

In Logistic Regression, the dependent variable is binary, meaning it possesses two distinct values such as True/False, 0/1, or Yes/No. This model serves to assess the likelihood of a dichotomous outcome based on one or more independent variables. Employing the logistic function, it establishes the relationship between the dependent variable and the independent variables.

Logistic Regression shares similarities with Linear Regression, as it stands as a special case within the realm of Generalized Linear Models. However, notable disparities exist between these two methodologies. In Logistic Regression, the Conditional Distribution y—x does not conform to a Gaussian distribution but rather adheres to a Bernoulli distribution. Predicted outcomes in Logistic Regression manifest as probabilities computed via the logistic function, constrained within the range of 0 to 1.

The outcomes derived from Logistic Regression can be expressed through the following equations:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n)}}$$

$$P(Y = 0|X) = 1 - P(Y = 1|X)$$

Where: - $P(Y = 1|X)$ represents the probability of the dependent variable Y being 1 given the values of the independent variables X. - $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ are the coefficients of the logistic regression model. - $X_1, X_2, \ldots, X_n$ are the independent variables. - e is the base of the natural logarithm. Hilbe (2009)

2.3 PCA-logistic regression

Principal Components Analysis (PCA) is a linear technique for dimensionality reduction, which means that it performs dimensionality reduction by embedding the data into a linear subspace of lower dimensionality. Although there exist various techniques to do so, PCA is by far the most popular (unsupervised) linear technique. Therefore, in our comparison, we only include PCA.

PCA constructs a low-dimensional representation of the data that describes as much of the variance in the data as possible. This is done by finding a linear basis of reduced dimensionality for the data, in which the amount of variance in the data is maximal.

In mathematical terms, PCA attempts to find a linear mapping M that maximizes the cost function trace$[M^\top \Sigma M]$, where Σ is the covariance matrix of the data.

The covariance matrix Σ is given by:

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^\top$$

where $\mathbf{x}_i$ represents the i-th data point and μ is the mean of the data.

van der Maaten, Postma, and van den Herik (2009)

3 Experiments

In this study, we carried out the following experiments:

1. **Regularization and Solver Comparison**: We Investigated the effect of different optimizer (solvers) on the performance of logistic regression models with various regularization techniques.
2. **Feature Selection via PCA**: We Experimented with different percentages of retained features using PCA (Principal Component Analysis) and observe how it impacts the model's accuracy.

3.1 Internet-of-things dataset

The RT-IoT2022 dataset provides a comprehensive collection sourced from an operational IoT infrastructure, encompassing a diverse range of IoT devices and intricate network attack scenarios. It encompasses both normal and adversarial behaviors, offering a realistic portrayal of real-world situations. The dataset features data from various IoT devices such as ThingSpeak-LED, Wipro-Bulb, and MQTT-Temp, alongside various simulated attacks. S. and Nagapadma (2024).

For the purpose of this study, we have re-framed the dataset for a binary classification problem, relabeling all IoT device traffic as 'Normal - 0' and simulated attacks as 'Attack - 1'. A concise overview of the dataset can be observed in the Table 1 below.

3.2 Experimental setup

To ensure the robustness of the study, we have set up our experiments in four distinct stages: Data Analysis, Pipeline Setup, Training, and Inference. Each stage plays a crucial role in ensuring the reliability and validity of our experimental findings.

3.2.1 Data Analysis

We thoroughly explore the dataset to gain insights into its features, including their types (numerical, categorical

Table 1: Dataset Overview

Dataset **Number of Features** **Instances**	RT-IoT2022 83 123117
Attack Labels (1)	'ARP_poisoning', 'DDOS_Slowloris', 'DOS_SYN_Hping', 'Metasploit_Brute_Force_SSH', 'NMAP_FIN_SCAN', 'NMAP_OS_DETECTION', 'NMAP_TCP_scan', 'NMAP_UDP_SCAN', 'NMAP_TREE_SCAN'
Normal Labels (0)	'MQTT_Publish', 'Thing_Speak', 'Wipro_bulb'

2) and distributions. Within the given dataset, we identified two categorical features in the input space, namely "protocol" and "service". On further examination for potential biases and discrepancies in the dataset we observed a potential bias towards attack traffic, which accounted for more than half of the dataset, same bias can be observed for protocol distribution. This finding sheds light on the imbalance in the dataset, which needs to be accounted before model can be trained.

Table 2: Data Types of Features

Data Type	**Count**
float64	47
int64	34
object	2

3.2.2 Data Splitting

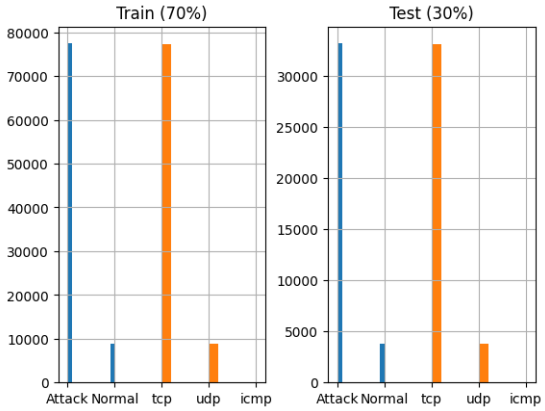

Figure 1: Train-Test Split

The dataset is divided into training and test sets, with the training set utilized for model training and the test set employed to assess the final model performance. To mitigate potential bias in the data distribution, we implemented a standard 70-30 split using StratifiedKFold, ensuring equal distribution of class labels across the dataset. This split strategy is illustrated in Figure 1.

3.2.3 Pipeline Setup

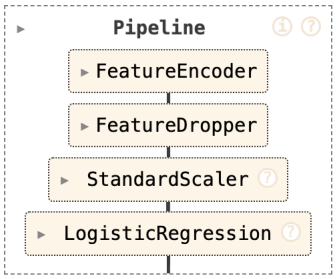

Figure 2: Pipeline

To ensure the consistency and efficiency of our experiments, we have established a systematic pipeline comprising several key steps.

1. **FeatureEncoder**: Categorical features are encoded into numerical representations. We utilize a custom encoder with one-hot encoding as its base to handle categorical variables effectively.

2. **FeatureDropper**: A custom feature dropper is implemented to remove redundant features such as port numbers and IDs.

3. **StandardScaler**: Features are standardized using the StandardScaler, which removes the mean and scales the data to unit variance.

3.3 Experiment 1: Regularization and Solver Comparison

This experiment aims to evaluate the impact of regularization techniques and compare different solvers for logistic regression models. The experiment is structured as follows:

1. **Model Setup**: Multiple logistic regression models are created using our initial pipeline (Figure 2).

2. **Regularization Techniques**: Each model's hyperparameters are fine-tuned with different penalties (None, l1, l2, elastic-net) using compatible solvers (refer to the compatibility matrix in Table 3).

3. **Performance Metrics**: Performance metrics including accuracy, precision, recall, and F1-score are computed for each model.

Table 3: Solver and Regularization Pairings

Penalty	lbfgs	liblinear	newton-cg	saga
None	✓		✓	✓
l1		✓		✓
l2	✓	✓	✓	✓
elastic-net				✓

Through this experiment, we aim to gain a deeper understanding of the role of regularization techniques in logistic regression models.

3.4 Experiment 2: Feature Selection via PCA

In this experiment, we investigate the impact of feature selection using Principal Component Analysis (PCA) on model performance. The experiment is structured as follows:

1. **Percentage of Retained Features**: We experiment with different percentages of retained features using PCA. By varying the number of principal components retained, we observe how it affects the model's accuracy and performance.

2. **Model Performance**: The performance of models trained with different percentages of retained features is evaluated using standard evaluation metrics such as accuracy, precision, recall, and F1-score.

Through this experiment, we aim to determine the optimal number of features to retain using PCA and its impact on model performance and generalization.

3.5 Results

3.5.1 Experiment 1 Result

In the Experiment 1 results, it's notable from Table 4 that L2 regularization with the Newton-CG solver achieved the highest accuracy of 98.6%. This indicates the effectiveness of this particular combination in optimizing the logistic regression model for the dataset.

Figure 3 showcases the performance of each solver in terms of accuracy. This visual representation helps to illustrate how different solvers compare in terms of their ability to optimize the model's accuracy.

Table 4: Performance Metrics

Parameters	Accuracy	Precision	Recall	F1
None-lbfgs	98.4	99.0	99.3	99.1
None-newton	92.3	92.9	99.0	95.9
None-saga	98.0	99.0	98.7	98.9
l1-liblinear	98.2	99.3	98.7	99.0
l1-saga	98.0	99.0	98.7	98.9
l2-lbfgs	98.4	99.0	99.2	99.1
l2-liblinear	98.2	99.1	98.8	99.0
l2-newton	98.6	99.4	99.1	99.2
l2-saga	98.0	99.0	98.7	98.9
elasticnet-saga	98.0	99.0	98.7	98.9

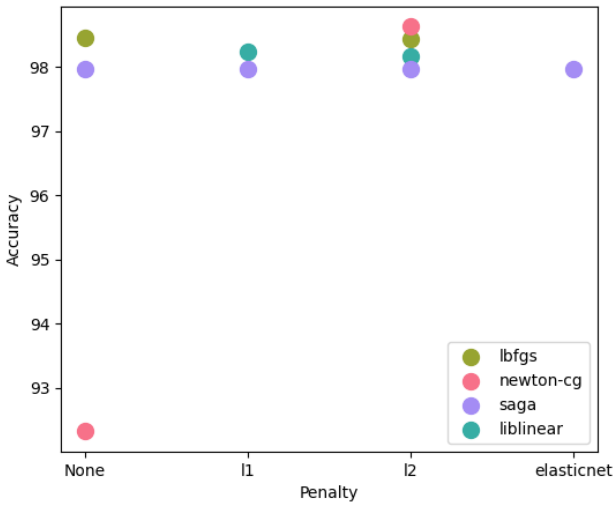

Figure 3: Regularization and Solver Comparison

3.5.2 Experiment 2 Result

In Experiment 2, we systematically explored the effect of increasing the dimensionality in our PCA-transformed feature set on model accuracy. Notably, there was a progressive improvement in accuracy correlating with the number of dimensions retained. The peak accuracy reached 98.8% when we preserved 59 dimensions, constituting about 75% of the original feature set. This observation indicates that the PCA was effective not only in reducing the dataset's complexity but also in retaining sufficient critical information to enhance the model's predictive accuracy.

Figure 4 visually depicts the relationship between the number of dimensions retained and the corresponding model accuracy, clearly illustrating the positive impact of increased dimensions up to a certain threshold on model performance.

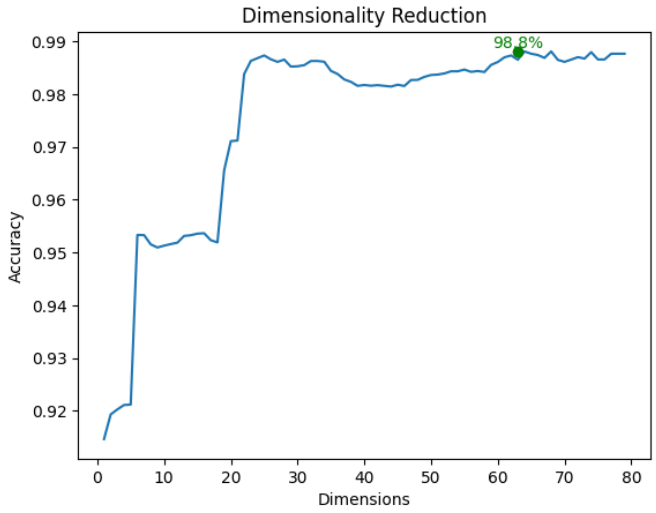

Figure 4: PCA Result

4 Discussion

1. **Effectiveness of Regularization Techniques**: Our findings reveal that L2 regularization paired with the Newton-CG solver achieved the highest accuracy of 98.6%. This underscores the efficacy of L2 regularization in managing overfitting, which is crucial in high-dimensional data environments like intrusion detection systems. L2 regularization's ability to penalize the square of coefficient values effectively constrains model complexity while maintaining predictive performance. Conversely, L1 regularization, which promotes sparsity by driving some coefficients to zero, demonstrated slightly lower performance. However, its utility in feature selection could be invaluable in scenarios where model interpretability is as critical as model performance.

2. **Impact of PCA on Model Performance**: The utilization of PCA to reduce dimensionality while retaining 75% of the original features resulted in a top accuracy of 98.8%. This finding illustrates that PCA can condense essential information into a reduced set of features without significantly compromising the model's accuracy. Such dimensionality reduction not only simplifies the model but also enhances computational efficiency, which is vital for real-time intrusion detection systems.

3. **Comparison with Other Studies**: Comparing our approach with the methodologies employed in prior studies such as those by Wang (2005) and Gu (2020), our logistic regression models appear to offer a balance between accuracy and computational demands. Unlike the multinomial logistic regression model by Wang et al., which targets specific attack types, our binary classification approach provides a broader application potential with high accuracy across various attack scenarios.

4.1 Theoretical and Practical Implications

The robust performance of logistic regression models equipped with L2 regularization and PCA highlights the robustness of these methods in cybersecurity applications, reinforcing the theoretical framework that supports logistic regression as a powerful tool for binary classification, particularly in environments as dynamic and vulnerable as IoT-based networks. Furthermore, the practical implications of our study are significant for cybersecurity professionals seeking efficient and accurate intrusion detection solutions. The demonstrated effectiveness of L2 regularization and PCA in reducing model complexity and improving prediction accuracy suggests that these techniques can be readily adopted in real-world intrusion detection systems, potentially enhancing their ability to thwart cyber-attacks.

4.2 Limitations and Future Research

While our results are promising, they are not without limitations. The potential biases in the RT-IoT2022 dataset and the generalizability of our findings to different network environments or attack types need further exploration. Future research could explore the integration of logistic regression with other machine learning techniques to address multi-class classification problems in intrusion detection. Additionally, examining the impact of other dimensionality reduction techniques could further enhance the applicability and effectiveness of logistic regression models in diverse cybersecurity contexts.

5 Conclusion

In conclusion, this study underscores the utility of logistic regression models in the field of cybersecurity, particularly for intrusion detection in IoT networks. The combination of L2 regularization and PCA not only achieves high accuracy but also maintains manageable model complexity, making it suitable for practical applications. The insights gained from this research contribute to the ongoing efforts to fortify digital infrastructures against evolving cyber threats.

References

Alan J., I. (2008). *Modern multivariate statistical techniques : Regression, classification, and manifold learning.* Springer. Retrieved from https://ezproxy.uow.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=275789&site=ehost-live

Gu, J. (2020). An effective intrusion detection model based on pls-logistic regression with feature augmentation. In *Cyber security: 17th china annual conference, cncert 2020, beijing, china, august 12, 2020, revised selected papers 17* (pp. 133–140).

Hilbe, J. (2009). *Logistic regression models* (1st ed. ed.). Boca Raton: Chapman Hall/CRC.

S., B., & Nagapadma, R. (2024). *RT-IoT2022* . UCI Machine Learning Repository. (DOI: https://doi.org/10.24432/C5P338)

Shyla, S., Bhatnagar, V., Bali, V., & Bali, S. (2022). Optimization of intrusion detection systems determined by ameliorated hnadam-sgd algorithm. *Electronics*, *11*(4), 507.

van der Maaten, L. J. P., Postma, E. O., & van den Herik, H. J. (2009). *Dimensionality reduction : A comparative review.* online. Retrieved March 2020, from https://lvdmaaten.github.io/publications/papers/TR_Dimensionality_Reduction_Review_2009.pdf

Wang, Y. (2005). A multinomial logistic regression modeling approach for anomaly intrusion detection. *Computers & Security*, *24*(8), 662–674.