AN AUTOML TOOL FOR SOFTWARE VULNERABILITY PREDICTION

TARIK GÜRCAN

AUGUST 2023

ÇANKAYA UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

M.Sc. Thesis in

COMPUTER ENGINEERING

AN AUTOML TOOL FOR SOFTWARE VULNERABILITY PREDICTION

TARIK GÜRCAN

AUGUST 2023

ABSTRACT

AN AUTOML TOOL FOR SOFTWARE VULNERABILITY PREDICTION

GÜRCAN, TARIK
M.Sc. in Computer Engineering

Supervisor: Assist. Prof. Dr. Gül TOKDEMİR
August 2023, 209 pages

The popularization of machine learning has been inevitable, especially in recent years, as large amounts of data have become more accessible and provide significant benefits for industries such as finance, healthcare, logistics, and technology. The need to implement machine learning applications, which have started to be used in different sectors, has increased rapidly and efficiently. However, the deep technical knowledge and coding skills required to build a machine-learning model using traditional methods may not be accessible to everyone. Therefore, no-code or low-code platforms have started to attract more attention. These platforms allow users to load datasets, define features for building the model, and even train and evaluate the model with non-technical interfaces. In this way, it eliminates the need to write complex code, making it possible for a wider audience to be involved in machine learning projects. In this thesis, a platform that enables performing no-code machine learning prediction models for software vulnerability analysis is developed, which offers important features such as run-time limiting and model degradation analysis in addition to other no-code platforms. This platform is named PPDaaS (Prediction Platform as a Service) and is used to train prediction models for “Software Vulnerability” and “OpenML” benchmarking datasets. Initially, the platform is tested with benchmarking data that is used for AutoML platforms. The performance of the classification algorithms is evaluated by placing a time limit on the "OpenML adult" dataset, to make a fair comparison between the machine learning algorithms. Once the

platform is tested for its intended AutoML functionalities, it is used to find model degradation on Software Vulnerability data. Finally, the feedback about the developed PPDaaS platform is gathered from end users as well. The results indicate that the developed PPDaaS platform can find a place for itself in the software industry.

Keywords: Machine Learning, Runtime Limit in Machine Learning, Model Degradation, No Code/Low Code Machine Learning, Software Vulnerability, Data Analysis

ÖZET

YAZILIM GÜVENLİK AÇIĞI TAHMİNİ İÇİN AUTOML ARACI

GÜRCAN, TARIK
Bilgisayar Mühendisliği Yüksek Lisans

Danışman: Dr. Öğr. Üyesi Gül TOKDEMİR
Ağustos 2023, 209 sayfa

Özellikle son yıllarda büyük miktarda verinin daha erişilebilir hale gelmesi ve finans, sağlık, lojistik ve teknoloji gibi sektörlere önemli faydalar sağlaması nedeniyle makine öğreniminin yaygınlaşması kaçınılmaz olmuştur. Farklı sektörlerde kullanılmaya başlayan makine öğrenmesi uygulamalarının hayata geçirilmesi ihtiyacı hızlı ve verimli bir şekilde artmaktadır. Ancak geleneksel yöntemleri kullanarak bir makine öğrenimi modeli oluşturmak için gereken derin teknik bilgi ve kodlama becerilerine herkes yetkin olmayabilir. Bu nedenden dolayı kodsuz veya az kodlu platformlar daha fazla ilgi görmeye başlamıştır. Bu platformlar, kullanıcıların veri kümelerini yüklemesine, modeli oluşturmaya yönelik özellikleri tanımlamasına ve hatta teknik olmayan arayüzlerle modeli eğitmesine ve değerlendirmesine olanak tanır. Bu sayede karmaşık kod yazma ihtiyacını ortadan kaldırarak daha geniş bir kitlenin makine öğrenimi projelerine dahil olmasını mümkün kılar. Bu tezde, diğer kodsuz platformlara ek olarak çalışma zamanı sınırlama ve model bozulma analizi gibi temel özellikleri sunan, yazılım zafiyet analizi için kodsuz makine öğrenimi tahmin modellerinin gerçekleştirilmesine olanak sağlayan bir platform geliştirilmiştir. Bu platform, PPDaaS (Prediction Platform as a Service) olarak adlandırılmıştır ve "Yazılım Güvenlik Açığı" ve "OpenML" veri kümeleriyle tahmin modellerini eğitmek için kullanılmıştır. Başlangıçta platform, AutoML platformları için kullanılan kıyaslama verileriyle test edildi. Sınıflandırma algoritmalarının performansı, makine öğrenimi algoritmaları arasında adil bir karşılaştırma yapmak için "OpenML yetişkin"

veri kümesine bir zaman sınırı konularak değerlendirildi. Platform, amaçlanan AutoML işlevleri açısından test edildikten sonra Yazılım Güvenlik Açığı verilerindeki model bozulmasını bulmak için kullanıldı. Son olarak geliştirilen PPDaaS platformuna ilişkin geri bildirimler son kullanıcılardan da toplandı. Sonuçlar geliştirilen PPDaaS platformunun yazılım sektöründe kendine yer bulabileceğini göstermektedir.

Anahtar Kelimeler: Otomatik Makine Öğrenimi, Makine Öğreniminde Zaman Sınırı, Model Bozunumu, Kodsuz/Düşük Kodlu Makine Öğrenimi, Yazılım Açığı, Veri Analizi

ACKNOWLEDGEMENT

First and foremost, I would like to extend my heartfelt gratitude to my advisor, Assist. Prof. Dr. Gül TOKDEMİR, for her exceptional guidance, ideas, and patience. Your guidance played a significant role in enhancing the quality of this thesis. Similarly, I would like to thank my family and girlfriend, Ms. Pelin Külah, who has been by my side throughout this process. Your patience, understanding, and unwavering support made me never feel alone during this challenging journey. Knowing that you were always there for me has been my most significant source of strength.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

SYMBOLS

Σ	: Uppercase Sigma (Summation Notation)

ABBREVIATIONS

PPDaaS	: Prediction Platform as a Service
Gaussian NB	: Gaussian Naïve Bayes
MLP	: Multi-Layer Perceptron
Bernoulli NB	: Bernoulli Naive Bayes
SVC	: Support Vector Classifier
SGD	: Stochastic Gradient Descent
XGB	: XG Boost
LGBM	: Light GBM
Ridge CV	: Ridge Cross-Validated
SVR	: Support Vector Regressor
LGBM	: Light GBM
XGB	: XG Boost
Nu SVR	: Nu Support Vector Regressor
RBF	: Radial Basis Function
PCA	: Principal Component Analysis
SVD	: Singular Value Decomposition
RFE	: Recursive Feature Elimination
RFECV	: Recursive Feature Elimination with Cross-Validation
ICA	: Independent Component Analysis
CSV	: Comma-Separated Values

CHAPTER I
INTRODUCTION

The ability of computers to learn from the data and to improve their performance without any programming has enhanced machine learning and artificial intelligence which has a significant effect on our lives [1]. In today's world, artificial intelligence, and machine learning change [2] many domains ranging from medical [3][4][5], finance [6][7][8], business [9], empirical sciences [1], technology [10], consumer services [1], identification of bugs [1] and logistics [1]. Therefore, applying machine learning (ML) techniques and extracting valuable insights from the data has become more important. It led to no and low code machine learning platforms [11][12][13][14] to emerge in the response to growing diversity of machine learning applications and cause the demand for machine learning solutions. This resulted in challenges associated with traditional manual model development. These platforms (e.g., Weka[15], H2O.ai[16]) enabled users to apply any model to his/her data without writing any code which increased the population of the data-science workers. Thus, automated machine learning tools (AutoML) have increasingly popular in today's world. However, there are very few AutoML platforms that provide runtime limits for experimentation with fair comparisons of different models. These tools work as applied machine learning systems that run ML algorithms for as much time as needed and evaluate the results. The absence of a runtime constraint in automatic machine learning tools gives rise to apprehensions regarding fairness and the feasibility of conducting unbiased comparisons between various ML algorithms. In today's world, the popularity, and the diversity in domains of machine learning applications have accelerated data accumulation which might affect the performance of the algorithms. As data is generated from the systems continuously, the previously created prediction models might not work properly on the newly generated datasets. Hence, the performance of the ML prediction models might degrade as new data arrives from resources. This might have a significant impact on the performance and reliability of the prediction models. Thus, the recognition of the potential for data degradation is

important for data scientists and software practitioners who heavily depend on predictive models to make well-informed decisions. Accordingly, in the software world, various machine learning models are utilized to enhance the quality of systems by detecting errors[17], defects[18], or vulnerabilities[19][21]. The software vulnerabilities are critical weaknesses that can expose systems to threats [22] and risks [23]. Particularly, detecting software vulnerability has become vital [24][25] when developing software systems [26] as it may cause attackers might use it to their advantage, which could compromise the system[27]. Accordingly, several software metrics were collected for the prediction of the vulnerabilities in C/C++ and Java-based systems[28]. Although metrics are utilized to identify vulnerabilities[20] through machine learning, prediction models built using data at different granularity levels (class and method level) have not been studied before [29] for the recently generated Software Vulnerability Datasets data.

The absence of the runtime limit option and model degradation analysis in AutoML systems shows that there is a need for an innovative and comprehensive tool including these features as well. Hence, the primary objective of the thesis is to develop a no-code platform named Prediction Platform as a Service (PPDaaS) that enables the user to apply machine learning models automatically with or without a given run time limit for fair comparison; apply feature selection models and observe model degradation over a period on software vulnerability data. Therefore, after the development of the platform, the software vulnerability data provided by Pereira and colleagues [28] in 2022 with all granularity levels namely, function, classes, and files are utilized for building prediction models. Studies of software vulnerability prediction are simulated by separating this dataset at the function granularity level and conducting experiments on it. In addition to that, a benchmark dataset from the openML platform is used in validating the AutoML platform. OpenML [30] is an open-source platform that provides machine learning datasets, algorithms, and experiments that allow researchers and data scientists to compare their works. Finally, the developed AutoML platform is tested by software professionals and their feedback is gathered through a survey instrument.

1.1 THESIS STRUCTURE

The rest of this thesis is organized as follows: Chapter 2 will include an in-depth literature review on the topics of machine learning and its diversity, software vulnerability, and automated machine learning tools. Chapter 3 will present the design of the architecture of the Platform named Prediction Platform as a Service (PPDaaS) and the features of this platform that show how a user can utilize and interact with it. Chapter 4 will explain the experimental setup and results of the experiments performed on the PPDaaS platform using the Software Vulnerability dataset and OpenML adult dataset. Chapter 5 will discuss the results of the experiments and draw an insight into the tool's strengths and limitations. Finally, Chapter 6 will conclude the thesis, and summarize the contributions and recommendations for future study to improve the effectiveness and usefulness of PPDaaS in practical settings.

1.2 THESIS OBJECTIVE

This thesis aims to bridge the gap in the field of automated machine learning tools by introducing the PPDaaS. By developing this tool, we strive to enable the researchers to apply a large set of machine learning models without writing any code use a large set of feature selection techniques with or without running specific time limits, and easily analyze model degradation over time on software vulnerability data. Accordingly, this thesis addresses four main research questions:

1. What are the missing features in AutoML tools?
2. How effective PPDaaS (Platform for Prediction as a Service) is?
3. How does software vulnerability prediction model performance change at different granularity levels?
4. How does prediction model performance change as new data arrive?

CHAPTER II
BACKGROUND

In this chapter, we will explore the importance of machine learning, its diverse applications, its integration into no-code/low-code platforms, and its role in addressing software vulnerabilities, thereby aiming to fill the existing gap in the literature.

2.1 EXPLORING THE SIGNIFICANCE OF MACHINE LEARNING

In the field of machine learning, many studies highlight the importance of machine learning. For example, Anute and colleagues [9] investigated the effects of machine learning on business organizations and operations. Furthermore, Sarker [10] provided a thorough overview of the machine-learning techniques that may be used to improve the functionality and intelligence of an application.

Jordan and colleagues [1], discuss the dramatic progression of machine learning applications over the past decades. It has become the approach of choice in AI for creating useful software for robot control, computer vision, and speech recognition which has a significant impact on computer science, and also on sectors that deal with data such as consumer services, identification of bugs, medicine, complex systems, and logistics. In addition to these, Jordan and colleagues [1] state that it has a big impact on empirical sciences including biology cosmology, and social science. Accordingly,

Gong and colleagues [2] state that the methods of machine learning have reached high performance and have been widely used in a variety of applications that are used in the real world. For example, Lee and colleagues [3] compare Gradient boosting, Support vector machine, Random Forest, Cat Boost, multilayered perceptron, and AdaBoost machine learning models to predict malaria by using patient data. They state that machine learning can have promising results. Gogas and Papadimitriou evaluate the machine learning effects on finance and economics[8]. The authors of [4] have reviewed current cardiology literature and reported on how doctors in the area are utilizing artificial intelligence tools, notably machine learning

approaches. The study of [5] has aimed to describe the usage of machine learning techniques on biological data to biologists. According to Goodell and colleagues [6] the rise of machine learning and artificial intelligence significantly affected the financial industry. Similarly, the study of [7] focuses on the effects of machine learning techniques on the financial sector and they evaluate techniques with traditional methods.

2.2 MACHINE LEARNING INTEGRATION IN NO-CODE AND LOW-CODE PLATFORMS

A study by Lebens and colleagues [11] explores the utilization of low-code and no-code development platforms to enhance an organization's development capabilities. The study revealed that companies of various sizes were utilizing these platforms, frequently in combination with workflow automation.

AutoML is an acronym for Automated Machine Learning that refers to the process of automating the development of machine learning algorithms with various features such as feature selection, data preprocessing, and model training. The study conducted by Hanussek M. and colleagues [12] discusses that automated machine learning can surpass humans. In addition, they study a comparative analysis of four AutoML frameworks using a selection of 12 widely recognized datasets sourced from OpenML. The datasets encompass a balanced distribution of both supervised classification and supervised regression tasks, with six datasets assigned to each category.

Musigmann M and colleagues [13] examine the viability, as well as the merits and drawbacks, of constructing AutoML models in comparison to constructing traditional machine learning models. They use a total of 20 distinct machine-learning algorithms that were concurrently trained and evaluated. To facilitate comparison, both a neural network and various conventional machine learning algorithms were trained and subsequently tested. The findings presented in the article demonstrate the promising potential of AutoML.

Eggensperger K el al. [14] provide a new solution for automated machine learning tools (AutoML). They provide Auto-Sklearn 2.0 as an AutoML system to run under the time limits named budget allocation. They evaluate their systems by datasets located on the openML website.

In Table 1, we observe key differences between various AutoML platforms and our platform PPDaaS. Generally, all platforms provide preprocessing, automatic execution of machine learning algorithms, and feature selection methods. Specifically, Auto-Sklearn 1.0 and Auto-Sklearn 2.0 provide runtime limits but are different than ours. Their approach differs from ours, we simulate all algorithms at the same time, and they eliminate algorithms that exceed a certain runtime. However, none of them provides a model degradation analysis. Therefore, our provided PPDaaS platform aims to fill these gaps, offering an enhanced runtime limit and model degradation feature.

Table 1: Comparison Table of the AutoML Tools

Platform	Preprocessing Capabilities	Machine Learning Algorithms	Feature Selection	Runtime Limit	Model Degradation Feature
Auto-Sklearn 1.0	Yes	Yes	Yes	Yes	No
Auto-Sklearn 2.0	Yes	Yes	Yes	Yes	No
H2O	Yes	Yes	Yes	No	No
Weka	Yes	Yes	Yes	No	No
PPDaaS	Yes	Yes	Yes	Yes (Improved Customized)	Yes

2.3 SOFTWARE VULNERABILITY DETECTION

Software vulnerability refers to flaws or weaknesses within a software system within a software system [19][21]. These vulnerabilities stem from errors in coding, design, or system architecture. A software vulnerability is vital because it makes the platform vulnerable to attacks or other unauthorized actions. A system's vulnerability may be defined as a weakness because it tends to be susceptibility to attack from external factors such as viruses, hackers, or others [22]. Hence, software security and software vulnerability have long become seen as vital when developing software [26], as they pose serious risks to security [23].

In recent years, vulnerability exposure has attracted significant attention [24] and machine learning approaches are crucial in accurate vulnerability predictions[25]. Accordingly, several researchers studied various machine-learning techniques for detecting software vulnerabilities [22][23].

Pereira [31] studied the integration of different ML techniques to enhance the performance of vulnerability detection. The article states that Static Code Analysis has a good detection ratio, but it may give many false positives. Another study by Pereira and colleagues [32] examined the automatic detection of vulnerabilities. They are focused on static techniques to detect vulnerabilities in C functions. They also processed all features by Deep Graph Convolution Neural Network (DGCNN). They found this technique enhances the recall but decreases the precision. Similarly, Abdulrahman and colleagues [33] study information about the effect of code smells on software security and evaluate open-source integration under the Tomcat Software developed by Apache. They concluded that the relation between vulnerability and smells has not significantly impacted each other.

A study by Alves H and colleagues [34] focuses on the relationship between the vulnerabilities and the software metrics. Their analysis depends on 2875 patches to build a metric and vulnerability dataset for the classes, functions, and files. The researchers performed calculations on software metrics derived from their data sources. They then applied a correlation algorithm and conducted statistical tests on these metrics to determine any potential relationships between them and the preexisting vulnerabilities. The results show that software metrics discriminate the non-vulnerability and vulnerable functions. However, it is not possible to state a significant relationship between the number of vulnerabilities and metrics.

An article by Mederios N and colleagues [35] conducted two experiments. In the first experiment, multiple vulnerability prediction models are built using software metrics. In the second experiment, a consensus decision-making strategy is proposed for classifying the code levels, based on the output of different classification models. The results indicate that the rate of false alarms does not allow us to use machine learning algorithms.

The authors of [28] provide a dataset comprising vulnerabilities obtained from five prominent open-source C/C++ projects, namely Mozilla, Linux Kernel, Xen, httpd, and Glibc. They developed a method to collect necessary data that relies on an automated process utilizing vulnerabilities sourced from the Common Vulnerability and Exposures (CVE) Details website. The researchers obtained the relevant source code units for each vulnerability from the project repository, including both the vulnerable and patched versions. Following this, the researchers performed a comprehensive analysis of Software Metrics (SMs) on the code levels, and they

conducted Static Analysis Tools to get security alerts. They specifically targeted potential vulnerabilities and weaknesses. They gathered the dataset consisting of a total of 5214 vulnerabilities. To demonstrate its practicality, the researchers investigated the application of the dataset for training machine learning models designed to identify susceptible C/C++ functions. The findings of the study indicate that the dataset has practical applications and is a significant contribution to the field of software security, particularly for researchers in this area.

The study of [36] focuses on the alternations in software metrics after the resolution of a specific vulnerability by the developer. They applied comparative analysis on a total of 250 files obtained from Apache Tomcat and Apache CXF. Similarly, Siavvas's [37] thesis introduces software measures (such as complexity metrics) for distinguishing between vulnerabilities and investigating the possibility of linkages between vulnerabilities. While the analysis showed that there may be major interdependencies between different types of vulnerabilities, it also showed that software metrics are only useful as weak indications of specific security risks.

The researchers of [38] study the impact of data balancing techniques on the efficacy of software vulnerability prediction models. The experimental design comprises four distinct balancing methods and seven different classification algorithms. In addition to these, they used text mining-based features. Their results show that to obtain accurate estimations of software components that are prone to vulnerabilities, it is necessary to balance highly unbalanced datasets used for software vulnerability prediction. The utilization of software metric features proves to be advantageous in accurately characterizing vulnerable samples, while the implementation of ensemble-based classifiers yields superior outcomes.

The study in [39] investigates various code patterns utilized in the identification of vulnerabilities and assesses their efficacy in comparison to conventional software metrics. The researchers have developed an adaptive vulnerability prediction model through the analysis of security vulnerabilities in Java web applications, Apache CXF, and Tomcat. Machine learning and statistical techniques are employed to predict vulnerabilities by utilizing metrics and traceable trends as features.

Gupta A. and colleagues [40] used a collection of software measures that were converted into understandable regulations through the application of machine learning. 32 supervised techniques for machine learning have been evaluated for the three most common software system vulnerabilities, such as the *LawofDemeter*,

BeanMemberShouldSerialize, and *LocalVariablecouldBeFinal*. In this study, the J48 machine learning method was applied, and the vulnerability detection accuracy was up to 96%. They stated that the results are verified using tenfold cross-validation, and they have also been analyzed using statistical metrics like the ROC curve, Kappa statistics, Recall, Precision, etc.

The authors [41] put forth a methodology for the automated assessment of the efficacy of static code analyzers, utilizing Common Vulnerabilities and Exposures (CVE) reports as a basis for evaluation. The researchers evaluated five open-source static C code analyzers, as well as one commercial alternative. The findings indicate that while static C analyzers have demonstrated strong performance in benchmark tests involving artificial bugs, contemporary tools fail to detect a significant portion, ranging from 47% to 80% of vulnerabilities present in a benchmark dataset consisting of real-world programs.

The researchers in [42] proposed a method to evaluate the performance of machine learning (ML) based solutions for detecting software vulnerabilities. They conducted three experiments, focusing on ML techniques for detecting software vulnerabilities in different scenarios. The first experiment examined ML techniques for detecting a single type of vulnerability, while the second explored ML techniques for detecting multiple types of vulnerabilities within a dataset. The third experiment investigated ML techniques for detecting software vulnerabilities when faced with class imbalance, using varying an imbalance ratio. The empirical findings demonstrate the feasibility of utilizing machine learning techniques for software vulnerability detection.

Shukla and colleagues [43] proposed a model that statistically evaluating in the software vulnerability area. They considered with severity of vulnerability. Their experiment results showed that estimating the necessary information becomes more accurate than now.

The study done by Bahaa and colleagues [21] considered how to use techniques of learning such as machine learning and deep learning to detect vulnerabilities. They achieved results that indicate that deep learning and machine learning obtained positive outcomes in the field of detecting software vulnerability. Similarly, Bassi and colleagues [44] observed the effect of hyperparameter optimization on Software Vulnerability Models. They were focused on evaluating the effect of dual hyperparameter optimizations on Software Vulnerability Models. Their results showed

that increasing the productivity of Software Vulnerability models with dual HPO is more than 64% effective.

Pereira and colleagues [45] proposed a technique for gathering software vulnerabilities from databases and improving them with static data which are SAT alarms and SMs. The article studied by Ugrenovic and colleagues [46] evaluated the pros and cons of the one-class Support Vector Machine, Local Outlier Factor, and Isolation Forest to detect the distribution of the data. Their results showed that Isolation Forest is a superior method to the others.

Ugrenovic and colleagues [47] studied different approaches based on a collection of one-class autoencoders. This framework can be capable of both picture classification and image detection. Experimental results demonstrate its robustness to unknown pictures while maintaining accuracy on par with the state of the neural networks for datasets.

The correlation between software metrics and the existence of vulnerabilities is studied by Fonseca and colleagues [48]. The analysis is based on 2875 security patches, used to build a dataset with metrics and vulnerabilities for all the functions, classes, and files of 5750 versions of five widely used projects that are exposed to attacks: Linux Kernel, Mozilla, Xen Hypervisor, httpd, and glibc. the software metrics from their sources are calculated and correlation algorithms and statistical tests are used on these metrics to identify relations between them and the existing vulnerabilities. In another study, Fonseca et al, [49] used a representative and large dataset the examine the state-of-the-art vulnerability prediction techniques. Their obtained results showed that the dataset can be used to distinguish.

Using data accessible in the National Vulnerability Database and other open-source repositories, Bhandari and colleagues [50] proposed the CVEfixes dataset and an automated collecting method to compile it.

In the study of [29], researchers focused on metrics at various granularity levels (methods and java classes). They applied machine learning techniques to predict vulnerability in classes or methods. Their results show that the class-level metrics suppress within the 70 higher in recall and 75% higher in a prediction of vulnerability.

Clemente and colleagues [51] compare the traditional machine learning methods of naive Bayes, decision trees, support vector machines, and random forests with the deep learning technique. Their findings indicate that deep learning methods generate higher performance in predicting software security bugs.

The article by Salimi and colleagues [52] focused on studies of software metrics to predict software vulnerability. They focused on if they change the metrics, will the model's performance increase or decrease. They introduced new metrics named 'Vulnerable Slices' and tested and compared them with metrics that predict the vulnerability. Their results show that the accuracy of the model had been increased with the new metrics that they provide.

In the study of Medeiros and colleagues [20], the authors examined the effects of software metrics in detecting software vulnerability. Their results state the software metrics assist in identifying and predicting vulnerability.

Siavvas and colleagues [53] shed light on the ability of the software metrics to predict software vulnerability among different types. Their results indicate that software metrics are not adequate for predicting specific types of vulnerability. Campos and colleagues [54] provide a framework named Propheticus. Their opinion was that tools can be used to decrease the complexity of using machine learning techniques.

Kaya et al [55] examined the impacts of the classifiers, feature types, and data balancing techniques on the performance of the predicting software vulnerability model. Their results state that, smaller datasets have good performance at Random Forest on the opposite larger datasets have good performance at Rusboost Tree. As seen from the literature, many studies were conducted on software vulnerability detection that incorporates various ML techniques. However, none of the studies utilize no/low code platforms that enable fair comparison of ML algorithms and provide analysis of model degradation when new data is obtained from resources.

CHAPTER III
RESEARCH METHODOLOGY

During this research, existing literature in the field of Software Vulnerability, Machine Learning, No Code Platforms like AutoML are explored. Through this exploration, we identified several features that are missing in AutoML platforms. Hence, a platform that offers Automatic No Code Machine Learning Model Building named Prediction Platform as a Service (PPDaaS) is developed that includes run-time limit and model degradation analysis features. In the section below, we will discuss the details of the platform.

3.1 PLATFORM PREDICTION AS A SERVICE PLATFORM

In this section, the details of the platform that provides an automatic machine-learning solution that can be utilized without writing any code are explained. This platform allows users to apply various machine learning techniques without knowledge of coding or details of algorithms.

For the development of the platform and execution of the experiments, the following system configurations and libraries were used:

Hardware Specifications:

- **Processor:** Intel Core i5-8250U 1.60GHz
- **Memory:** 8GB RAM

Software and Libraries:

- **Programming Languages:** Python 3.7 for backend and Vue 2.6 for frontend
- **Backend Libraries:**
    - Flask for creating the web service
    - Scikit-learn (sklearn) for machine learning algorithms
    - XGBoost for gradient boosting framework
    - NumPy for numerical operations
    - Pandas for data manipulation

- **Frontend Libraries:**
    - XLSX for handling Excel files
    - Axios for making HTTP requests

The users can utilize the platform by dragging/dropping a CSV file into the interface and then the journey begins. Users can specify the random state and test size of their model by using the interface. The following sections will explain the details of the platform features.

Random State: The random state parameter is used for the control of the level of randomness in actions or algorithms. Specifying the random state in the machine learning algorithm enables users to gain more reliable outcomes. In our platform, the default value of random state is set to 40. Users can change the random state in a text box. If the user has not set the random state, it will use the default value.

Test Size: The test size refers to the subset of the dataset for evaluating the performance measurements of the trained model. In the machine learning process, users should create two data sets named trained and test sets. The test set is used for the evaluation of the trained model performance without applying any training process to it. Test size parameters are commonly stated as percentages. For example, a 40% test size means that 40% of the data is reserved for testing and the remaining is reserved for the training process. In our platform, the default value has been set to 0.4. Users can enter the values in the range of 0 to 1 that set the test size of a train test split.

PPDaaS provides a large set of classification and regression algorithms including Decision Tree Classifier, Random Forest Classifier, Gaussian Naïve Bayes, Linear Regression, Decision Tree Regressor, Random Forest Regressor, etc. as shown in Table 2 and Table 3. PPDaaS enables users to select these algorithms for the uploaded dataset. The platform automatically calculates all algorithm's results at once and gives the result to the user. It makes the user easily see which algorithm is more preferrable for the dataset.

Table 2: Classification Algorithms Provided by the PPDaaS Platform.

Classifiers			
Decision Tree Classifier	Gaussian NB (Gaussian Naive Bayes)	Bagging Classifier	Linear Discriminant Analysis
Random Forest Classifier	Gradient Boosting Classifier	Extra Tree Classifier	Quadratic Discriminant Analysis
KNeighbors Classifier	MLP Classifier (Multi-Layer Perceptron Classifier)	Bernoulli NB (Bernoulli Naive Bayes)	Hist Gradient Boosting Classifier
SVC (Support Vector Classifier)	Ada Boost Classifier	SGD Classifier (Stochastic Gradient Descent Classifier)	Stacking Classifier
Voting Classifier	Label Spreading	XGB Classifier (XG Boost Classifier)	LGBM Classifier (Light GBM Classifier)
Self Training Classifier	Categorical NB	Nearest Centroid	Label Propagation

Table 3: Regression Algorithms Provided by the PPDaaS Platform

Regressors			
Linear Regression	KNeighbors Regressor	Bayesian Ridge	Ridge CV (Ridge Cross-Validated)
Ridge	SVR (Support Vector Regressor)	Extra Tree Regressor	LGBM Regressor (Light GBM Regressor)
Lasso	Gradient Boosting Regressor	Hist Gradient Boosting Regressor	XGB Regressor (XGBoost Regressor)
Elastic Net	MLP Regressor (Multi-Layer Perceptron Regressor)	Multi-Output Regressor	Stacking Regressor
Decision Tree Regressor	Ada Boost Regressor	NuSVR (Nu Support Vector Regressor)	Bagging Regressor
Random Forest Regressor			

PPDaaS platform provides a very large set of feature selection techniques to enhance users selecting the feature selection technique to their own needs. Moreover, it enables the application of all feature selection models to data separately. The results of feature selection with any of the machine learning algorithm is listed. The list of feature selection techniques includes PCA, SelectKBest, RFE, SparsePCA, etc. as shown in Table 4.

Table 4: All Feature Selection Methods Provided by The Platform

Feature Selection Methods			
SelectKBest with f_classif	Select From Model with Lasso	RFE with Logistic Regression	Sparse PCA
SelectKBest with mutual_info_classif	SelectFromModel with Ridge	RFE with SVC	Mini Batch Sparse PCA
SelectKBest with chi2	SelectFromModel with Elastic Net	RFECV with estimator Random Forest Classifier	Truncated SVD
SelectKBest with f_regression	RFE with Decision Tree Classifier	RFECV with estimator Logistic Regression	Fast ICA
Sequential Feature Selector (SVC),	RFE with Random Forest Classifier	PCA	RFECV with estimator SVC
Sequential Feature Selector with Random Forest Classifier,	RFE with estimator MLP Classifier	Kernel PCA	RFECV with estimator MLP Classifier
Sequential Feature Selector with Gradient Boosting Classifier,	RFECV with estimator Decision Tree Classifier		

3.2 RUNTIME LIMIT FEATURE

The PPDaaS platform allows users to set a limit on the duration of the machine learning process through the Runtime Limit feature. This functionality is based on two parameters which are time and batch size. Users can set the maximum time for the runtime and batch size. In the process, first, all data rows are randomized by the platform. The runtime limit works by fitting the model with the first batch size and controlling the time if it exceeds or not. If not, the second batch size will be operated, and the cycle will repeatedly proceed on other batches. This feature allows a user can make reliable comparisons between different machine learning algorithms. This makes users easily compare each machine-learning algorithm results within the same execution time.

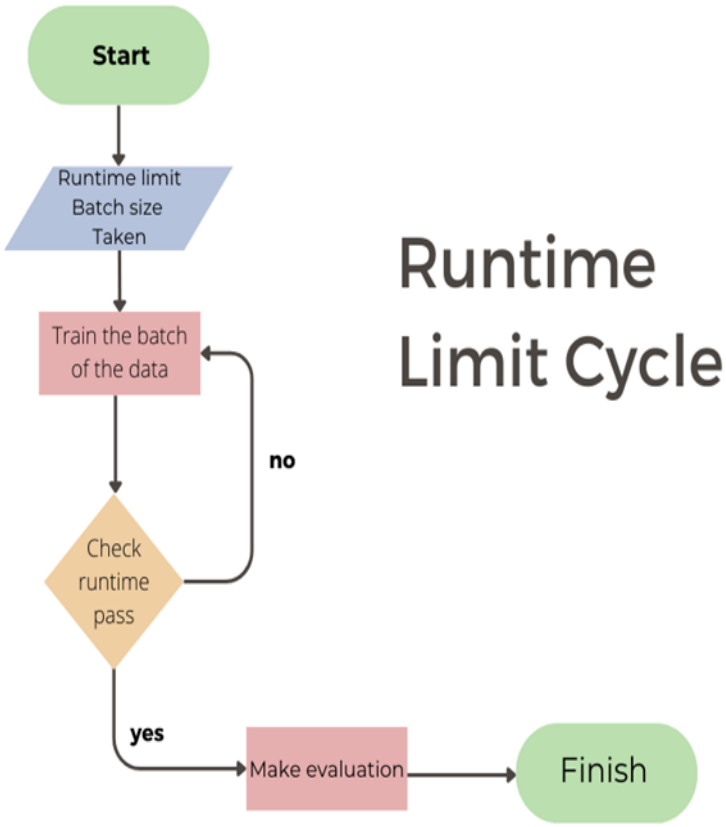


Figure 1: Runtime Limit Cycle

3.3 MODEL DEGRADATION FEATURE

PPDaaS platform allows users to analyze their machine-learning results over a period as their data accumulates. For example, a user has data about COVID-19 in 2021-2022. By using PPDaaS, performance score results are taken and after a year, new data has been added to the previous data. After that, the new results can be taken from the platform. In that use case, the model degradation feature can help users compare the results of two machine-learning applications. The tool shows the model performance differences between the datasets.

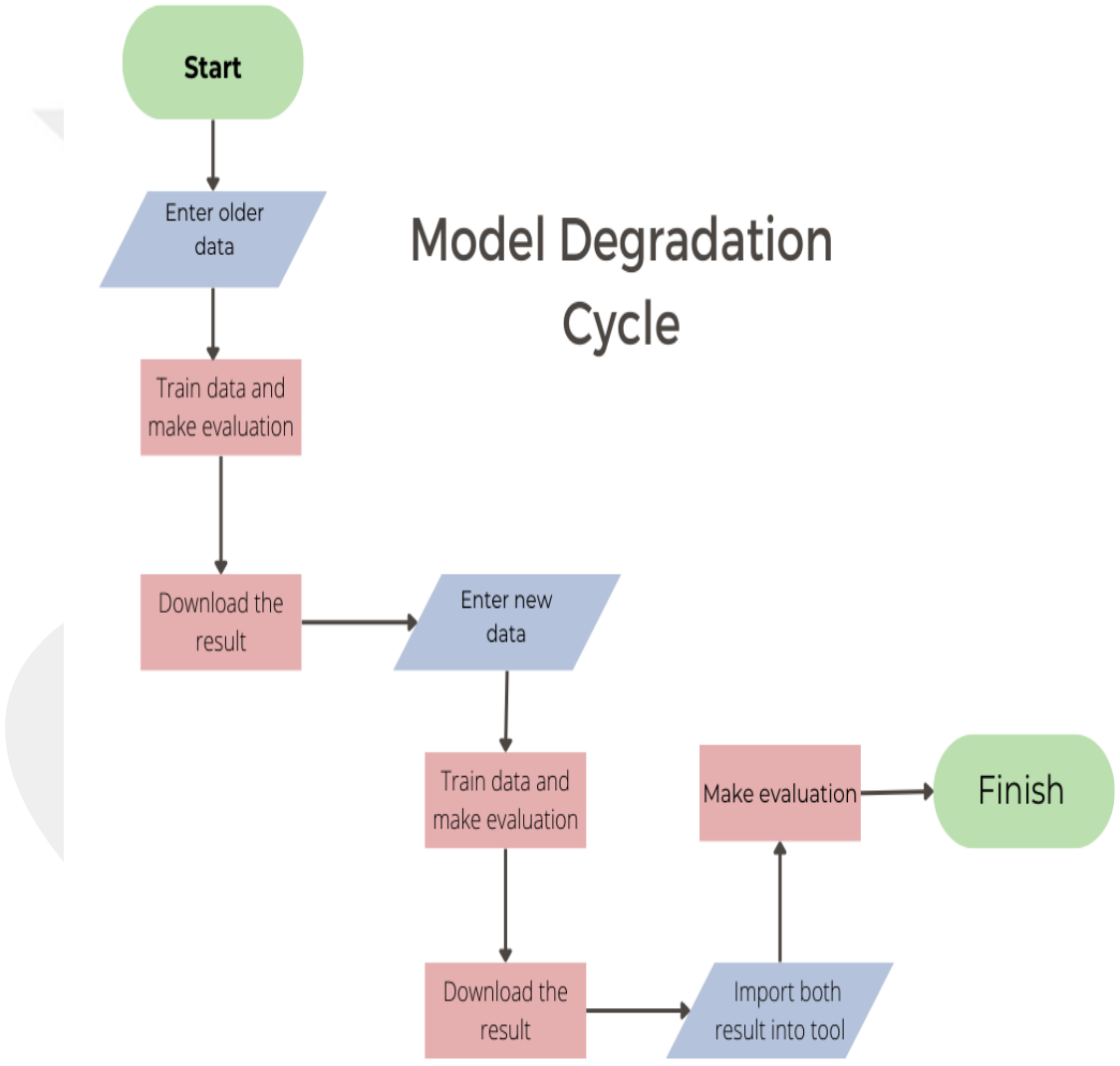


Figure 2: Model Degradation Cycle

3.4 INTERFACES OF THE PPDAS

3.4.1 Import Data

The dataset to be used for the analysis can be uploaded by dragging/dropping the CSV file into the box located on the platform. The drag-drop feature reduces the importing data complexities and makes the platform more user-friendly. The main page of our platform is shown in Figure 3.

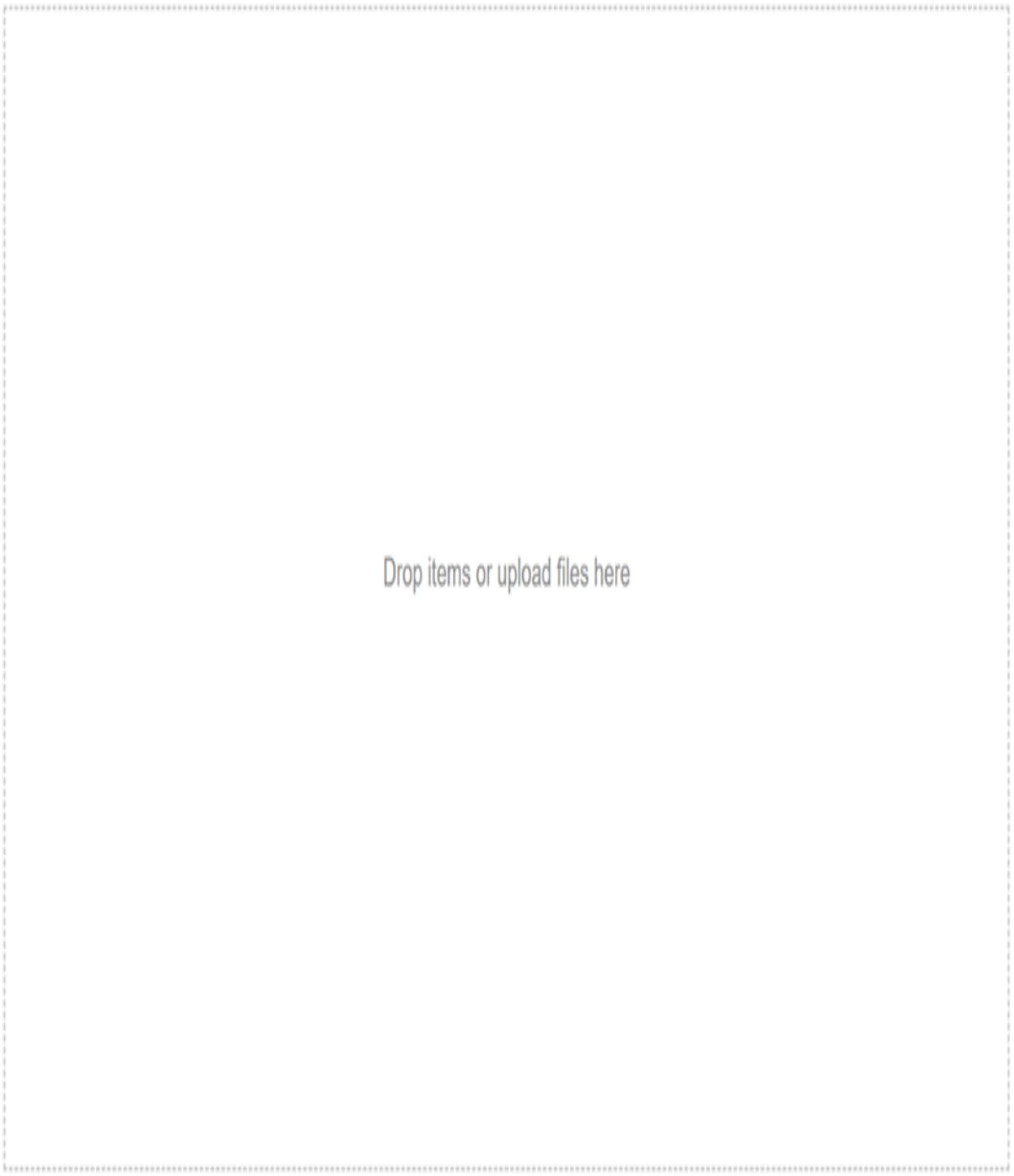


Figure 3: Main Page of The Platform

3.4.2 Specifying the Main Options

After the data import stage is completed, the tool displays a drag-and-drop options window. This page allows the user to specify the options by dragging items into a page or entering values into a text box. That step includes the user to customize parameters related to machine learning algorithms including random state, test size, and a runtime limit.

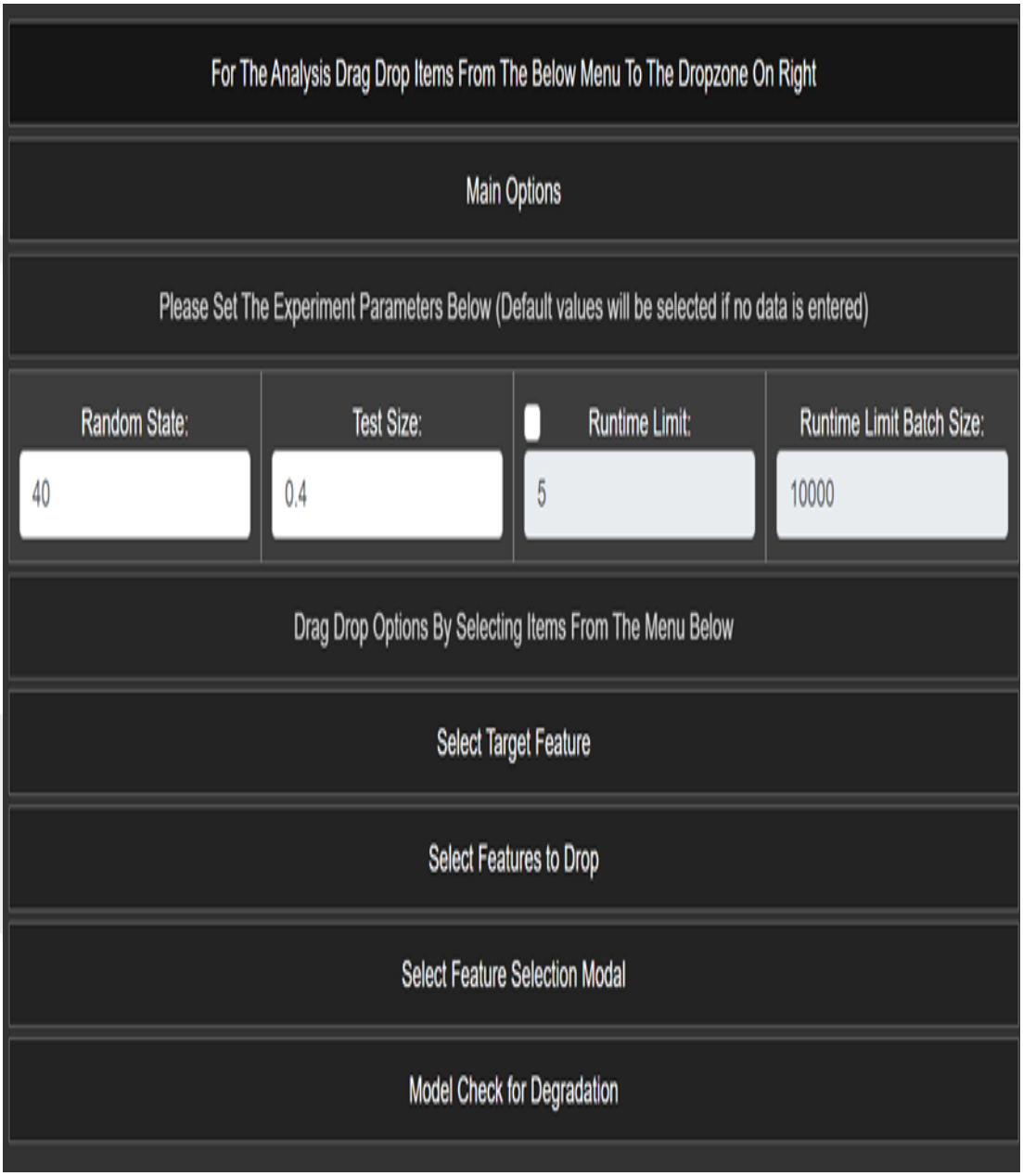


Figure 4: Options Section of The Platform

3.4.3 Target Feature Selection

By clicking the select target feature tab located on the right pane, the accordion menu opens with the feature names of the data. Using the drag/drop feature, all features of the dataset are shown on the left pane of the platform below the Select Target Feature Tab. Users can specify the target feature by drag/ drop feature.

Figure 5: Target Feature Selection Section

3.4.4 Drop Features

During the preprocessing step, the user may want to drop some irrelevant features from the dataset. With the drag/drop feature, the user can easily drop features by dragging the feature onto the drop box of the platform located on the right side of the screen.

Figure 6: Drop Feature Section

3.4.5 Feature Selection Algorithms

At this step, the user has already selected the main options, target feature, and features to be dropped. Hence, the user can select the feature selection method and select the feature count for the machine learning algorithm. Users can select just one or multiple feature selection methods. In addition to that, users can select all feature selection methods at once. Each feature selection method is used by each machine learning algorithm as a result.

Figure 7: Feature Selection Section

3.4.6 Label Encoding

If the dataset contains categorical data, the label encoder comes into play. Label encoder turns categorical data such as words or names, it turns these words into numbers. For example, when confronted with categories, say, "Small", "Medium", or "Large", this tool gracefully translates them into a numerical format think 0, 1, and 2.

3.4.7 Building the Machine Learning Process

Once the previous options are selected, users can submit the dataset and the options to the platform. Platform starts to evaluate machine learning and feature selection algorithms at the backend side. The user will see the loading screen at the end of the calculation.

3.4.8 Result

After the machine learning process is completed, the tool provides an analysis result as a downloadable table. The table consists of columns of model type, accuracy/mean squared error, precision/root mean squared error, recall/mean absolute error, F1 score/R-squared, test size, random state, batch size, runtime limit, drop columns, selected column, and feature selection model.

3.4.9 Download Output

PPDaaS tool provides users with the capability to download the analysis result as a table. This feature allows for further examination of the data or presentation of the findings to relevant stakeholders.

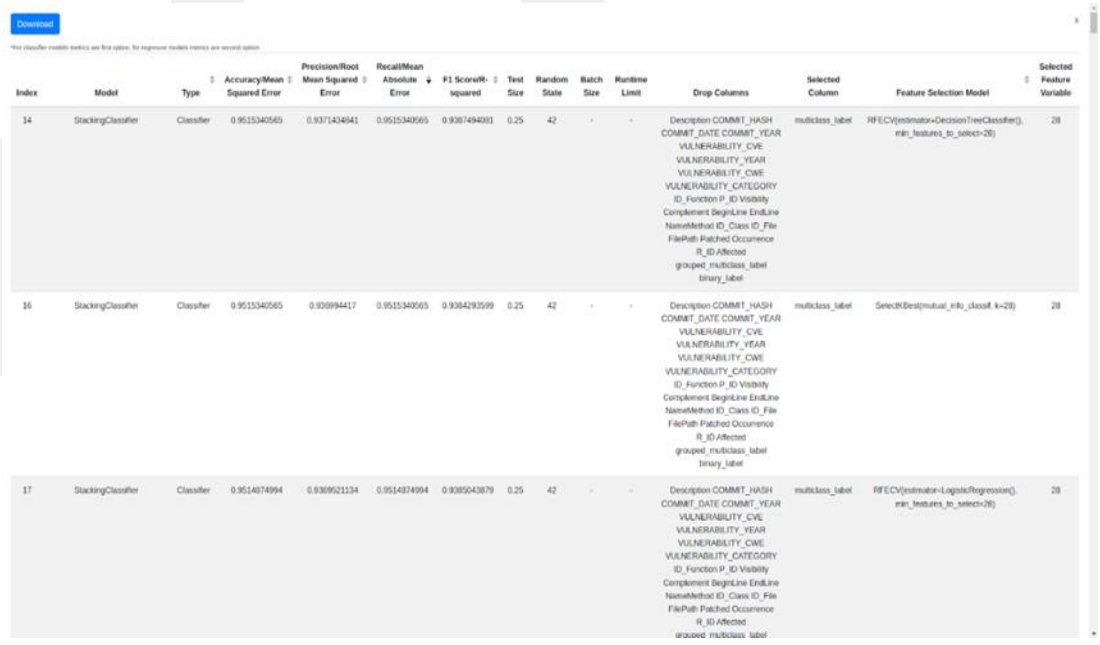


Figure 8: Model Performance Modal of the Platform

3.4.10 Model Degradation Analysis

After all the processing is finished, the tool provides a feature that enables comparison of the analysis results. The task can be done by comparing two machine learning analysis results which are previously taken from the platform. After a while new data might be added to the previously analyzed dataset and the results obtained for the new dataset and the previously obtained results could be compared for analyzing model degradation. This feature shows the stability and reliability of the machine learning models as data gets larger in time.

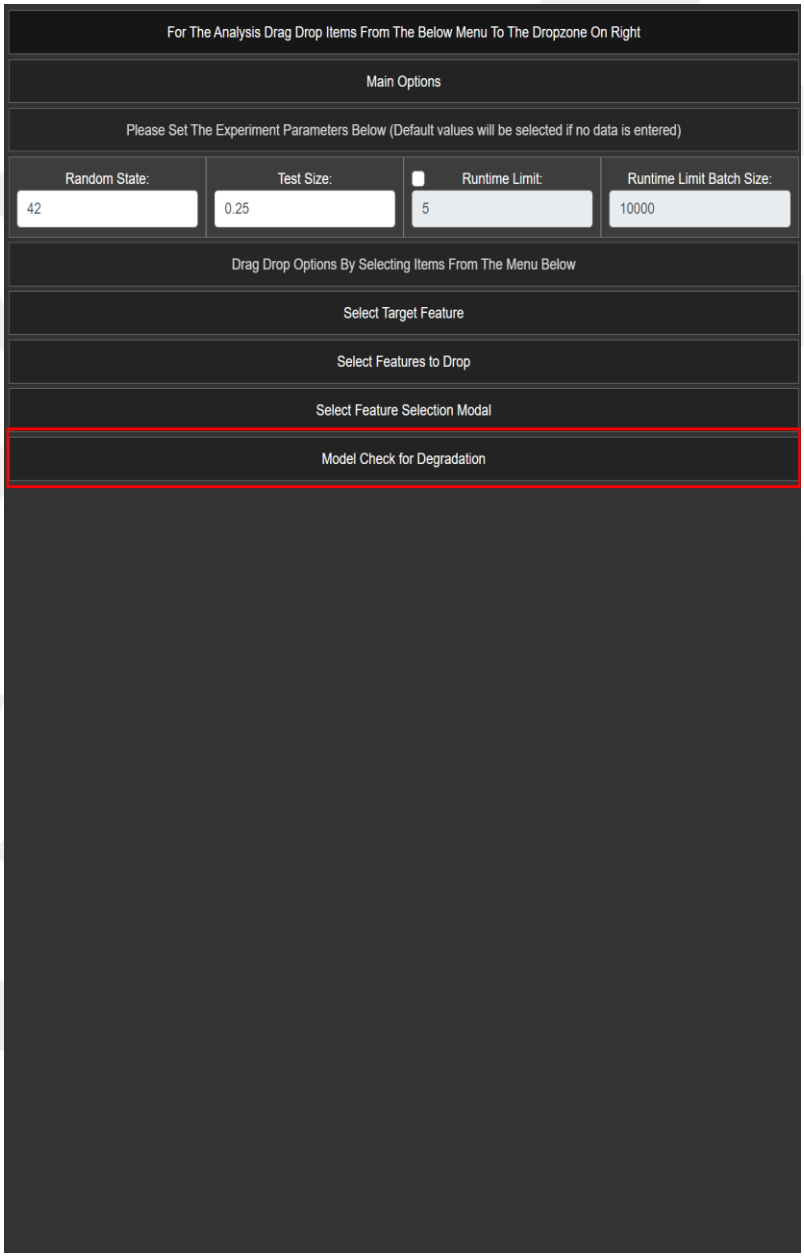


Figure 9: Model Degradation Tab

3.4.11 Model Performance Metrics Provided by PPDaaS Platform

The applied machine learning algorithms are evaluated based on various performance metrics such as Accuracy, Mean Squared Error, Precision Root Mean Squared Error, Recall, Mean Absolute Error, F1 Score, and R-squared. All evaluation metrics are explained below.

3.4.11.1 Accuracy

Accuracy is a metric used for evaluating classification models that measure the proportion of classified data correctly over the amount of data in the dataset.

Accuracy = TP + TN / TP + TN + FP + FN

3.4.11.2 Mean Squared Error (MSE)

Mean Squared Error is used for evaluating regression models that measure the average squared difference between the predicted and actual values.

MSE = (1/n) * Σ (y_pred - y_actual)^2

3.4.11.3 Precision

Mean Squared Error is used for evaluating regression models that measure the average squared difference between the predicted and actual values.

MSE = (1/n) * Σ (y_pred - y_actual)^2

3.4.11.4 Root Mean Squared Error (RMSE)

The RMSE quantifies the typical difference between the model's predictions and the observed data. In mathematical terms, it is the residuals' standard deviation. The residuals are the gaps between the data points and the regression line.

RMSE = √(1/n) * Σ (y_actual - y_predicted)^2

3.4.11.5 Recall

Recall, also known as True Positive Rate or Sensitivity, is a metric used in statistics and machine learning to assess how well a model can accurately identify all relevant occurrences within a dataset.

Recall = (True Positives) / (True Positives + False Negatives)

3.4.11.6 Mean Absolute Error (MAE)

Absolute error is the difference between a prediction and its true value. MAE measures group errors by averaging absolute errors from predictions and observations. The L1 loss function is MAE.

MAE = (1/n) * Σ|y_pred - y_actual|

3.4.11.7 F1 Score

The F1 score is a metric for the evaluation of models in machine learning that indicates how accurate they are. It brings together a model's precision and recall scores in a single value.

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

3.4.11.8 R-Squared(R^2)

R^2 is a statistical measure to assess how well a regression model fits the data. R-squared can take on any value from zero to one. If the data are a perfect fit for the model, then the projected value and the actual value will be identical, and the R-squared value will equal 1.

R-squared = 1 - (Sum of Squares of Residuals) / (Total Sum of Squares)

3.4.12 Survey About the Usability of PPDaaS

To collect the opinions of professionals about the PPDaaS platform, a usability study is performed. Several demo videos were prepared that show how the platform works, so everyone can understand it. The videos were shown with both verbal and multichoice survey options to software professionals and their opinions were gathered. Afterward, to create the survey questions, an initial draft was generated using zero-shot prompting [56] within Baize LLM [57], with version baize-v2-7b running as a demo under HuggingFace[58]. Then, the AI-generated survey questions and choices were manually edited for improvement. The details of the prompts and the survey questions used are given in Appendix 1.

Then the videos and the survey are shared with software professionals and their feedback is analyzed to gain insights into the usefulness and effectiveness of the PPDaaS platform.

CHAPTER IV
EXPERIMENTS AND RESULTS

The developed platform is used to analyze two datasets. Initially, the dataset used for benchmarking AutoML platforms is used to test the features of the system. Afterward, software vulnerability data is analyzed to see how model degradation can be analyzed. After the experiments, the platform is tested by potential users to gather feedback about its usability and effectiveness. Below the details of the experiments are explained.

4.1 DATA OVERVIEW

4.1.1 Benchmarking AutoML Data

PPDaaS (Prediction Platform as a Service) platform is a powerful platform that enables the application of a large set of machine learning algorithms and feature selection techniques at once efficiently. Thus, to test the capabilities of this platform, we used a benchmarking dataset used for AutoM platforms from the Open Machine Learning service for the first experiment.

OpenML is a service named Open Machine Learning, a free public software service for machine learning research. Service provides sharing datasets, experiments, and algorithms that are done by researchers and practitioners. In addition to these, OpenML is a platform where researchers and practitioners can compare their results with others.

The dataset obtained from the OpemML platform is named an adult dataset, which is used to analyze how an individual's annual income is influenced by a multitude of factors. The factors that influence it include the individual's education level, age, gender, occupation, and other relevant variables. This dataset is frequently referenced in K-Nearest Neighbors (KNN) literature. The dataset features can be seen in Figure 10.

	id	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class
0	1	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	2	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	3	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	4	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	5	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	48838	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
48838	48839	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
48839	48840	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
48840	48841	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
48841	48842	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

Figure 10: OpenML Data Overview

The attribute income has two category levels: Individuals with an annual income of less than or equal to $50,000 and those with an annual income exceeding $50,000. The number of attributes is 14. The other features of the dataset such as age, education, or occupation are the demographic characteristics and additional attributes utilized to delineate an individual. The potential for predicting an individual's income level based on their personal information can be investigated.

4.1.2 Vulnerability Experiment Data

The software vulnerability dataset is "The dataset, which was created and made openly available on GitHub by Pereira and colleagues [28]. For this dataset, they created an automated process for building a dataset build with vulnerability metadata, including SMs and SAT Alerts. The steps for retrieving data as listed below.

The first step is to retrieve reported vulnerabilities from the online platforms. They started the first step by querying the CVE Details website to be able to take the list of published vulnerabilities for chosen projects. For the Vulnerability data, they considered and saved the metadata consisting of.

- Common Vulnerabilities and Exposures (CVE) identifier: A unique identifier that is assigned to each vulnerability.
- Common Vulnerability Scoring System (CVSS) score: The severity of the vulnerability based on its environment and characteristics. •impact on confidentiality, integrity, and availability
- Exploit difficulty
- Authentication Requirement
- Zero or more vulnerability types
- Common Weakness Enumeration (CWE) maps a vulnerability to a specific weakness.

The second step was, that they used commit hashes which are gained from collecting the metadata of the vulnerabilities, to be able to collect the vulnerable version and the normal version for each. They focused on Git repositories to be able to find the code version of the vulnerability and the correct version before patching. In short, they have two types of affected files, which are vulnerable and non-vulnerable (after patching).

In the third step, Pereira, and colleagues [28] used a third-party tool to perform SCA in C++ code levels which are function file classes, and they extracted SAT alerts and Software Metrics. They used two security analysis tools (SATs) named Flawfinder3 v2.0.10 and Cppcheck4 v1.82 to produce security alerts. These tools are used to take the files as input and give the output as a CSV file.

After all these in the last step, they collected all the vulnerability metadata and generated the SAT alerts and software metrics for the vulnerable and non-vulnerable data.

The creation of a large dataset includes Mozilla (i.e., the source code for the Firefox browser), the Linux Kernel, the Xen Hypervisor, the Apache HTTP Server (httpd), and the GNU C Library (Glibc). That dataset contains projects with different complexities, security needs, and sizes. For instance, 21.6 million lines of code came from Linux Kernal, and 24.6 million lines of code came from Mozilla. In contrast, others were much smaller than comparing xen, httpd, and glibc.

To be able to apply machine learning algorithms they applied simple SQL queries that slice and dice the data. So that the dataset that we use in our thesis has been created. It consists of two features as follows.

- Various levels of code units which means the different granularity levels correspond to different organizational units of the software. For example, functions files and classes.
- The number of alert types per function in the SAT system varies, as each SAT can generate zero or more alerts for each specific alert type. The number of alerts per type was extracted for each function to identify potential security issues associated with the code.

The dataset comprised a total of 126 different SAT alert kinds from Flawfinder and a collection of 28 software metrics for each function class or file. All features have been shown in the Appendix 2. The target class is considered as multiclass classification. This represents each function, file or class is vulnerable with a non-defined category or vulnerable in any category.

4.2 VULNERABILITY DATASET EXPERIMENT

The PPDaaS is configured for the Vulnerability dataset that we discussed in the previous section. We applied experiments on various levels of data granularity namely functions, class, and files.

4.2.1 Functions Code Level

To be able to observe the model degradation of the dataset in a period, we separated the data according to vulnerability years as shown in Figure 11.

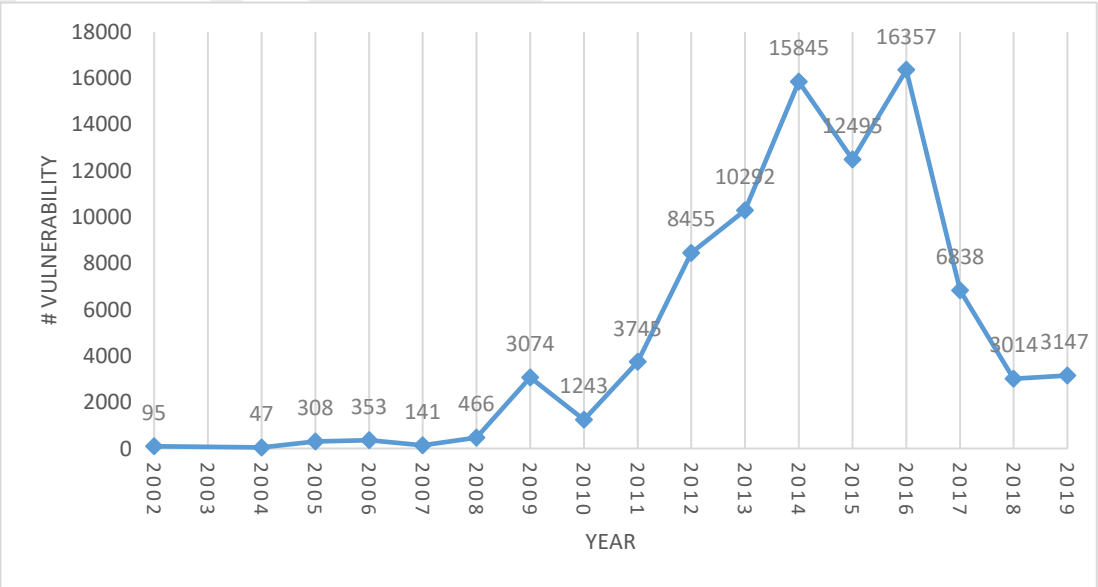


Figure 11: Reported Vulnerabilities Concerning Year

Figure 11 shows that the reported vulnerabilities started low in 2002, before a significant increase in 2005, and decreased in 2004. Until 2014, it caught an increasing trend and reached 15845. Before reaching the highest point, it slightly decreases before 2016. After that, it continues the alternating periods of decrease and slight increase. Thus, we separated the data into 3 parts based on the vulnerability count. in the years 2002-2014, 2002-2016, and 2002-2019 as shown in Table 5.

Table 5: Vulnerability Count in Each Period

2002-2014	2002-2016	2002-2019
44064	72916	85915

The random state is set to 42 and the test size is set to 0.25. Target (Class) Feature set to multiclass label similar to the Pereira and colleagues [28] and dropped columns are listed in Appendix 2. We drop these features because of they are some of these are unique identifiers that do not provide any meaningful information and some of these are metadata that do not contribute to the target prediction.

To observe the effect of various feature selection methods on this dataset and to reduce its dimensions, I applied different combinations of feature selection. After evaluating these combinations, I determined that selecting 28 features on the platform struck an optimal balance between training efficiency and model accuracy.

4.2.2 Class Code Level and File Code Level

We applied the same experiment settings as the Function Code Level settings on the PPDaaS. To be described briefly, we set the random state value as 42 and the test size as 0.25. Our evaluation has been conducted on the multiclass_label feature. The feature selection count option on the platform is set to 28, which is used to determine which feature selection approach is best for the data and to apply feature selection methods for lowering the dataset's dimensions.

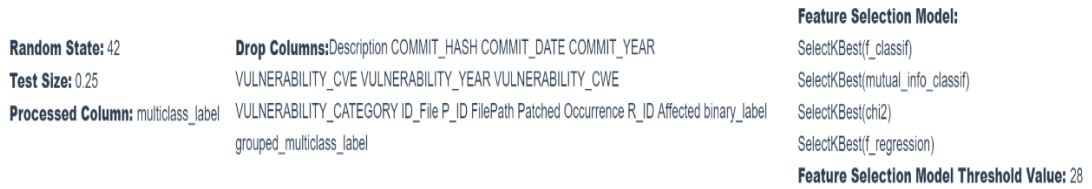


Figure 12: Class Code Level Settings Display in Platform

4.3 BECHMARKING AUTOML EXPERIMENT

The PPDaaS platform is configured for the OpenML adult dataset that we discussed in the previous section in the Dataset part. The platform is configured with a random state of 40.

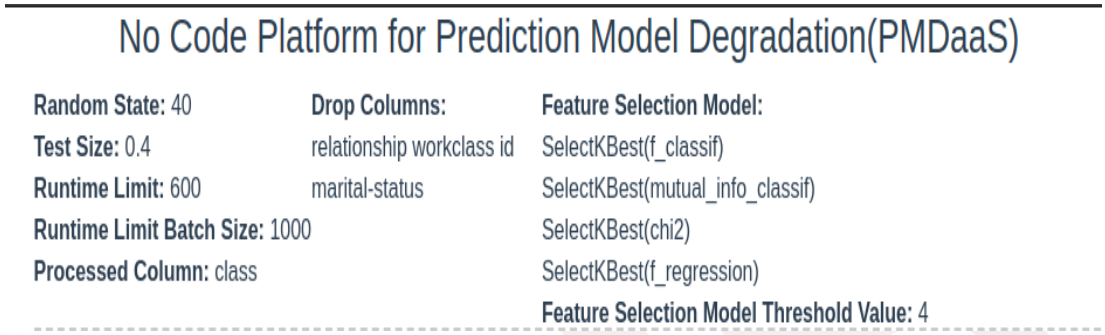


Figure 13: Optimized Values Display in Platform

To be able to split data, the platform configured the test size with 0.4. In addition to these, the platform serves runtime limit and runtime batch size to make fair comparisons. We set the runtime limit as 10 minutes same as [14]. Therefore, we can do a benchmarking analysis on our system by comparing other autoML tools. Drop columns are selected as id, relationship, marital status, and work class because id values do not carry any meaningful information, relationship, and marital status might have multicollinearity and occupation might give a more detailed or relevant insight about the work rather than work class. Lastly, we selected all feature selection models to be applied to the data, and to observe the effect of various feature selection methods on this dataset and to reduce its dimensions, I applied different combinations of feature selection. After evaluating these combinations, I determined that selecting feature selection counts 4 and the target feature as a class. Figure 14 shows that the dataset has a total of nine categorical variables work class, relationship, marital status, education, race, sex, occupation, native country, and class.

	id	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class
0	1	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	2	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	3	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	4	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	5	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	48838	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
48838	48839	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
48839	48840	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
48840	48841	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
48841	48842	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

Figure 14: Overview of The OpenML Adult Data

In this experiment, we integrate the label encoder into our platform which uses the label encoder for encoding categorical data to numerical data. This applied to all categorical data of the dataset which are education, race, sex, occupation, and native country. Label encoder is a data preprocessing technique that is used for converting categorical data to unique numerical values. After the machine learning process has been finished, the testing dataset is used to evaluate the model to test its performance. To determine how well the model can predict the class attribute, evaluation metrics such as precision, recall, F1-score, and ROC-AUC are computed.

4.4 VULNERABILITY EXPERIMENT RESULTS

4.4.1 Function Code Level

4.4.1.1 Results Between the Years 2002 to 2014

The top results of the function code level experiment between 2002 to 2014 according sorted by Accuracy, Precision, Recall, and F1 Score are shown in Table 6, Table 7, Table 8, and Table 9. All the results are listed in Appendix 3.

Table 6: Results of the 2002-2014 Period Sorted by Accuracy

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Stacking Classifier	0.950436	0.938478	0.950436	0.939923	Sparse PCA
Random Forest Classifier	0.950436	0.938336	0.950436	0.939579	PCA
Random Forest Classifier	0.950254	0.938236	0.950254	0.939873	Mini Batch Sparse PCA
Stacking Classifier	0.950254	0.938117	0.950254	0.939617	Mini Batch Sparse PCA
Stacking Classifier	0.950163	0.937936	0.950163	0.939464	PCA

Table 7: Results of the 2002-2014 Period Sorted by Precision

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
SVC	0.947712	0.940082	0.947712	0.923982	SelectKBest with f_classif
SVC	0.947712	0.940082	0.947712	0.923982	SelectKBest with mutual_info_classif
SVC	0.947712	0.940082	0.947712	0.923982	SelectKBest with f_regression
SVC	0.947712	0.940082	0.947712	0.923982	RFE with estimator Decision Tree Classifier
SVC	0.947712	0.940082	0.947712	0.923982	RFE with estimator Random Forest Classifier

Table 8: Results of the 2002-2014 Period Sorted by Recall

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Stacking Classifier	0.950436	0.938478	0.950436	0.939923	Sparse PCA
Random Forest Classifier	0.950436	0.938336	0.950436	0.939579	PCA
Random Forest Classifier	0.950254	0.938236	0.950254	0.939873	Mini Batch Sparse PCA
Stacking Classifier	0.950254	0.938117	0.950254	0.939617	Mini Batch Sparse PCA
Stacking Classifier	0.950163	0.937936	0.950163	0.939464	PCA

Table 9: Results of the 2002-2014 Period Sorted by F1 Score

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Stacking Classifier	0.950436	0.938478	0.950436	0.939923	Sparse PCA
Random Forest Classifier	0.950254	0.938236	0.950254	0.939873	Mini Batch Sparse PCA
Stacking Classifier	0.950254	0.938117	0.950254	0.939617	Mini Batch Sparse PCA
Random Forest Classifier	0.950436	0.938336	0.950436	0.939579	PCA
Stacking Classifier	0.950163	0.937936	0.950163	0.939464	PCA

4.4.1.2 Results Between the Years 2002 to 2016

The top results of the function code level experiment between 2002 to 2016 according sorted by Accuracy, Precision, Recall, and F1 Score are shown in Table 10, Table 11, Table 12, and Table 13. All the results are listed in Appendix 4.

Table 10: Results of the 2002-2016 Period Sorted by Accuracy

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.957143	0.94753	0.957143	0.947079	PCA
Random Forest Classifier	0.957086	0.94745	0.957086	0.947523	RFE with estimator MLP Classifier
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	SelectKBest with mutual_info_classif
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	RFE with estimator SVC
Stacking Classifier	0.956914	0.947139	0.956914	0.946003	RFE with estimator SVC

Table 11: Results of the 2002-2016 Period Sorted by Precision

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.957143	0.94753	0.957143	0.947079	PCA
Random Forest Classifier	0.957086	0.94745	0.957086	0.947523	RFE with estimator MLP Classifier
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	SelectKBestwith mutual_info_classif
RandomForestClassifier	0.957029	0.94733	0.957029	0.94732	RFE with estimator SVC
StackingClassifier	0.956914	0.947139	0.956914	0.946003	RFE with estimator SVC

Table 12: Results of the 2002-2016 Period Sorted by Recall

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.957143	0.94753	0.957143	0.947079	PCA
Random Forest Classifier	0.957086	0.94745	0.957086	0.947523	RFE with estimator MLP Classifier
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	SelectKBest with mutual_info_classif
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	RFE with estimator SVC
Stacking Classifier	0.956914	0.947139	0.956914	0.946003	RFE with estimator SVC

Table 13: Results of the 2002-2016 Period Sorted by F1 Score

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.957086	0.94745	0.957086	0.947523	RFE with estimator MLP Classifier
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	SelectKBest with mutual_info_classif
Random Forest Classifier	0.957029	0.94733	0.957029	0.94732	RFE with estimator SVC
Random Forest Classifier	0.956629	0.946666	0.956629	0.947231	SelectKBest with f_regression
Random Forest Classifier	0.957143	0.94753	0.957143	0.947079	PCA

4.4.1.3 Results Between the Years 2002 to 2019

The top results of the function code level experiment between 2002 to 2019 sorted by Accuracy, Precision, Recall, and F1 Score are shown in Table 14, Table 15, Table 16, and Table 17. All the results are listed in Appendix 5.

Table 14: Results of the 2002-2019 Period Sorted by Accuracy

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Stacking Classifier	0.951068	0.936839	0.951068	0.937856	Sparse PCA
Stacking Classifier	0.951068	0.936765	0.951068	0.937641	Mini Batch Sparse PCA
Stacking Classifier	0.951022	0.936595	0.951022	0.937392	RFECV with estimator SVC
Random Forest Classifier	0.950975	0.937009	0.950975	0.938623	PCA
Stacking Classifier	0.950929	0.936359	0.950929	0.937219	RFE with estimator Random Forest Classifier
Stacking Classifier	0.950882	0.936118	0.950882	0.936743	RFECV with estimator Decision Tree Classifier

Table 15: Results of the 2002-2019 Period Sorted by Precision

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
SGD Classifier	0.949253	0.951828	0.949253	0.924633	RFECV with estimator Random Forest Classifier
Random Forest Classifier	0.950975	0.937009	0.950975	0.938623	PCA
Stacking Classifier	0.951068	0.936839	0.951068	0.937856	Sparse PCA
Random Forest Classifier	0.950743	0.936812	0.950743	0.93886	SelectKBest with mutual_info_classif
Stacking Classifier	0.951068	0.936765	0.951068	0.937641	Mini Batch Sparse PCA
Random Forest Classifier	0.950696	0.936706	0.950696	0.938778	SelectKBest with f_regression

Table 16: Results of the 2002-2019 Period Sorted by Recall

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Stacking Classifier	0.951068	0.936839	0.951068	0.937856	Sparse PCA
Stacking Classifier	0.951068	0.936765	0.951068	0.937641	Mini Batch Sparse PCA
Stacking Classifier	0.951022	0.936595	0.951022	0.937392	RFECV with estimator SVC
Random Forest Classifier	0.950975	0.937009	0.950975	0.938623	PCA
Stacking Classifier	0.950929	0.936359	0.950929	0.937219	RFE with estimator Random Forest Classifier
Stacking Classifier	0.950882	0.936118	0.950882	0.936743	RFECV with estimator Decision Tree Classifier

Table 17: Results of the 2002-2019 Period Sorted by F1 Score

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.950743	0.936812	0.950743	0.93886	SelectKBest with mutual_info_classif
Random Forest Classifier	0.950696	0.936706	0.950696	0.938778	SelectKBest with f_regression
Random Forest Classifier	0.950975	0.937009	0.950975	0.938623	PCA
Random Forest Classifier	0.950556	0.936413	0.950556	0.938579	RFE with estimator SVC
Bagging Classifier	0.949718	0.935522	0.949718	0.938573	RFECV with estimator Decision Tree Classifier
Random Forest Classifier	0.950649	0.936513	0.950649	0.938545	RFECV with estimator Random Forest Classifier

4.4.2 Results of the Class Level

The top results of the class code level experiment sorted by Accuracy, Precision, Recall, and F1 Score are shown in Table 18, Table 19, Table 20, and Table 21. All the results are listed in Appendix 6.

Table 18: Results Sorted by Accuracy

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest with f_classif
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest with mutual_info_classif
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest with f_regression
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	RFE with estimator Decision Tree Classifier
RidgeClassifier CV	0.948179	0.932614	0.948179	0.928172	RFE Random Forest Classifier

Table 19: Results Sorted by Precision

Model	Accuracy	Precision	Recall	F1 Scor	Feature Selection Model
Random Forest Classifier	0.947347	0.936937	0.947347	0.940376	RFECV with estimator SVC
Bagging Classifier	0.946722	0.936175	0.946722	0.939751	RFE with estimator Random Forest Classifier
Bagging Classifier	0.947138	0.936016	0.947138	0.939561	SelectKBest with mutual_info_classif
Random Forest Classifier	0.946722	0.935988	0.946722	0.939587	RFECV with estimator Decision Tree Classifier
Bagging Classifier	0.946514	0.935986	0.946514	0.939597	RFECV with estimator SVC

Table 20: Results Sorted by Recall

Model	Accuracy	Precision	Recall	F1 Scor	Feature Selection Model
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest with f_classif
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest mutual_info_classif
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	SelectKBest with f_regression, k=28
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	RFE with estimator Decision Tree Classifier
Ridge Classifier CV	0.948179	0.932614	0.948179	0.928172	RFE with estimator Random Forest Classifier

Table 21: Results Sorted by F1 Score

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.947347	0.936937	0.947347	0.940376	RFECV with estimator SVC
Bagging Classifier	0.946722	0.936175	0.946722	0.939751	RFE with estimator Random Forest Classifier
Bagging Classifier	0.946514	0.935986	0.946514	0.939597	RFECV with estimator SVC
Random Forest Classifier	0.946722	0.935988	0.946722	0.939587	RFECV with estimator Decision Tree Classifier
Bagging Classifier	0.947138	0.936016	0.947138	0.939561	SelectKBest with mutual_info_classif

4.4.3 File Code Level Results

The top results of the file code level experiment according to sorted by Accuracy, Precision, Recall, and F1 Score are shown in Table 22, Table 23, Table 24, and Table 25. All the results are listed in Appendix 7.

Table 22: File Code Level Results Sorted by Accuracy

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
XGB Classifier	0.954795	0.944824	0.954795	0.936433	SelectKBest with f_classif
XGB Classifier	0.954795	0.944824	0.954795	0.936433	SelectKBest with mutual_info_classif
XGB Classifier	0.954795	0.944824	0.954795	0.936433	SelectKBest with f_regression
XGB Classifier	0.954795	0.944824	0.954795	0.936433	RFE with estimator Decision Tree Classifier
XGB Classifier	0.954795	0.944824	0.954795	0.936433	RFE with estimator Random Forest Classifier

Table 23: File Code Level Results Sorted by Precision

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Hist Gradient Boosting Classifier	0.953683	0.950056	0.953683	0.932078	RFECV with estimator Random Forest Classifier
Hist Gradient Boosting Classifier	0.954599	0.949567	0.954599	0.934694	RFE with estimator MLP Classifier
Hist Gradient Boosting Classifier	0.953749	0.948622	0.953749	0.932355	SparsePCA
Hist Gradient Boosting Classifier	0.954337	0.947664	0.954337	0.934204	SelectKBest with f_regression
Voting Classifier	0.953814	0.947612	0.953814	0.932631	RFECV with estimator Random Forest Classifier

Table 24: File Code Level Results Sorted by Recall

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
XGB Classifier	0.954795	0.944824	0.954795	0.936433	SelectKBest with f_classif
XGB Classifier	0.954795	0.944824	0.954795	0.936433	SelectKBest with mutual_info_classif
XGB Classifier	0.954795	0.944824	0.954795	0.936433	SelectKBest with f_regression
XGB Classifier	0.954795	0.944824	0.954795	0.936433	RFE with estimator Decision Tree Classifier
XGB Classifier	0.954795	0.944824	0.954795	0.936433	RFE with estimator Random Forest Classifier

Table 25: File Code Level Results Sorted by F1 Score

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
Random Forest Classifier	0.953356	0.936699	0.953356	0.937493	RFECV with estimator SVC
Random Forest Classifier	0.952963	0.935617	0.952963	0.937335	RFE estimator SVC
Random Forest Classifier	0.95316	0.936026	0.95316	0.937184	SelectKBest with f_regression
Random Forest Classifier	0.95316	0.935981	0.95316	0.937091	RFE with estimator Random Forest Classifier
Random Forest Classifier	0.952963	0.935344	0.952963	0.936874	RFE with estimator Decision Tree Classifier

4.5 BENCHMARKING AUTOML EXPERIMENT RESULTS

The top results of the benchmarking autoML experiment according to sorted by ROC AUC, Precision, Recall, and F1 Score are shown in Table 26, Table 27, Table 28, and Table 29. All the results are listed in Appendix 8.

Table 26: Results of the OpenML Data Experiment Sorted by ROC AUC

Model	ROC AUC	Precision	Recall	F1 Score	Feature Selection Model
Passive Aggressive Classifier	0.736446	0.464739	0.749367	0.57369	RFECV with estimator SVC
SGD Classifier	0.723537	0.524789	0.629747	0.572497	RFE with estimator MLP Classifier
SGD Classifier	0.717709	0.490028	0.653165	0.559957	RFECV with estimator Random Forest Classifier
SGD Classifier	0.715404	0.386526	0.876371	0.53645	RFECV with estimator Logistic Regression
SGD Classifier	0.711569	0.497151	0.625949	0.554165	RFECV estimator Decision Tree Classifier

Table 27: Results of the OpenML Data Experiment Sorted by Precision

Model	ROC AUC	Precision	Recall	F1 Score	Feature Selection Model
Passive Aggressive Classifier	0.592519	0.80084	0.201055	0.321417	RFECV with estimator MLP Classifier
Passive Aggressive Classifier	0.638139	0.684023	0.324262	0.43996	SelectKBest with mutual_info_classif
Gaussian NB	0.610679	0.671706	0.262447	0.377427	SelectKBest with f_classif
Gaussian NB	0.610679	0.671706	0.262447	0.377427	SelectKBest with mutual_info_classif
Gaussian NB	0.610679	0.671706	0.262447	0.377427	SelectKBest with f_regression

Table 28: Results of the OpenML Data Experiment Sorted by Recall

Model	ROC AUC	Precision	Recall	F1 Score	Feature Selection Model
SGD Classifier	0.715404	0.386526	0.876371	0.53645	RFECV with estimator Logistic Regression
Passive Aggressive Classifier	0.736446	0.464739	0.749367	0.57369	RFECV with estimator SVC
Passive Aggressive Classifier	0.682597	0.414557	0.666878	0.511282	Mini Batch Sparse PCA
SGD Classifier	0.717709	0.490028	0.653165	0.559957	RFECV with estimator Random Forest Classifier
SGD Classifier	0.699235	0.462627	0.634599	0.535136	PCA

Table 29: Results of the OpenML Data Experiment Sorted by F1 Score

Model	ROC AUC	Precision	Recall	F1 Score	Feature Selection Model
Passive Aggressive Classifier	0.736446	0.464739	0.749367	0.57369	RFECV with estimator SVC
SGD Classifier	0.723537	0.524789	0.629747	0.572497	RFE with estimator MLP Classifier
SGD Classifier	0.717709	0.490028	0.653165	0.559957	RFECV with estimator Random Forest Classifier
MLP Classifier	0.704278	0.610123	0.513713	0.557783	RFE with estimator Random Forest Classifier
SGD Classifier	0.711569	0.497151	0.625949	0.554165	RFECV with estimator Decision Tree Classifier

4.6 SURVEY RESULTS OF THE USABILITY OF PPDAAS

The platform is tested by users to get insights about its applicability and usage in real-world scenarios.

We created a video demonstrating the capabilities of the platform and asked software professionals with machine-learning backgrounds to watch the video and evaluate the system. We wanted to collect comments about the effectiveness and challenges of our platform. Therefore, initially, 10 software professionals were reached to get initial views on the platform. Then, an initial draft of the survey instrument using zero-shot prompting [57]within Baize LLM [57], with version baize-v2-7). Then, the AI-generated survey questions and choices were manually edited for improvement. The survey questions, accompanied by their respective answers, are presented below alongside their corresponding in Table 30. The participation in the survey has been diverse, ranging from academic individuals to those with long-term experience in the industry. Based on the survey, the age of the participants predominantly falls between 25 to 44 years. Specifically, four of them are aged between 25 and 34, while five are between 35 and 44. In terms of sectoral experience, most of the participants have over 5 years of experience, though there is also a notable number with experience ranging from 2 to 4 years. Considering this information, it can be deduced that the demographic primarily consists of mid-aged professionals with significant experience in the sector.

Table 30: Results of the Evaluation

Experience In Sector	5+	1-2	2-4	5+	2-4	5+	2-4	5+	5+	2-4
Age	35-44	25-34	25-34	45-54	25-34	35-44	25-34	35-44	35-44	25-34
Satisfaction Level	5	4	5	3	4	4	5	5	5	5
Understandability	4	4	4	3	3	3	3	5	5	4
User-friendliness	5	5	4	3	4	3	3	5	4	4
Effectiveness	3	5	5	4	3	4	4	5	4	5
Utility	3	4	3	4	4	4	5	5	4	5
Recommendation Likelihood	5	5	5	4	4	4	5	5	3	4
Future Usage Likelihood	4	4	5	3	4	4	5	5	3	4
Enjoyment Level	5	5	3	3	3	3	5	5	4	3
Learning Level from Video	4	4	5	2	3	2	5	4	5	5
Time-saving Potential	5	5	4	3	4	5	4	4	5	5
Quality Improvement Potential	4	4	4	3	4	4	4	4	5	5
Cost-saving Potential	5	5	5	4	4	5	4	5	5	5

CHAPTER V
DISCUSSION

The results of the experiments which are stated in chapter IV will be discussed below.

5.1 DISCUSSION ON OPENML ADULT DATA EXPERIMENT

The highest performance scores according to the evaluation metric have been shown below in Table 31.

Table 31: Highest Scores Concerning Score Parameters

Score Parameter	Score Value	Model	Feature Selection Method
ROC AUC	0.736446	Passive Aggressive Classifier	RFECV with estimator SVC
Precision	0.80084	Passive Aggressive Classifier	RFECV with estimator MLP
Recall	0.876371	SGD Classifier	RFECV with estimator Logistic Regression
F1 Score	0.57369	Passive Aggressive Classifier	RFECV with estimator SVC

The Passive Aggressive Classifier achieved the highest ROC_AUC value, it achieved around 0.736 with the feature selection technique RFECV with an estimator SVC with use as hyperparameter n components equals 4. The result shows that the model has a strong discriminatory capacity in differentiating positive and negative instances. In addition to this SGD Classifier obtained promising results with 0.723 which uses the RFE feature selection technique with an estimator MLP Classifier and hyperparameter tuning with set feature selection count 4.

The Passive Aggressive Classifier shows the highest precision among the classifiers with a value of around 0.80 with the feature selection RFECV with an estimator MLP Classifier and hyperparameter tuning as features to select as four. If we look closer at the top 5 precision values, it can be easily seen that Passive Aggressive Classifier and Gaussian Naïve Bayes have the highest values with different feature selection techniques. The SGD Classifier demonstrates the highest recall value

around 0.876 with the feature selection RFECV estimator Logistic Regression and used hyperparameter tuning technique with features to select as 4. This suggests that the model demonstrated efficacy in identifying a substantial portion of true positive instances, although it may have also generated a higher number of false positives, resulting in a reduced precision score.

The Passive Aggressive Classifier with RFECV with an estimator SVC and tuning as a feature to select equals 4 feature selection method obtained the highest value of F1 score with the value of 0.5736. This shows that the data have better performance metrics on Recall AUC and Precision. Therefore, it shows that the data is imbalanced.

By comparing the results on [14] their results based on 1-AUC and ours are ROC AUC, Precision, Recall, and F1 Score. Our lowest 1-AUC results equal around 0.263. With comparing the [14] results. Our results tend to approach their results. Their provided autoML solution platform obtained 0.1547 and ours was 0.263 shown in Table 32.

Table 32: Comparison Table with AutoML vs. PPDaaS

Platform	**1-AUC**	**Precision**	**Recall**	**F1 Score**
PPDaaS	0.2635	0.80084	0.876371	0.57369
Feurer and colleagues	0.1547	Not Provided	Not Provided	Not Provided

To sum up, we wanted to examine the performance of classifiers within a specific time to make a fair comparison by limiting the iterations by setting runtime limits. With the analysis results, Although, the Passive Aggressive Classifiers perform better than the others with a feature selection method such as RFECV with different estimators, The study of Thapa S [59] found the Random Forest Classifier to be their top-performing algorithm.

5.2 SOFTWARE VULNERABILITY EXPERIMENT

The results of the performance of machine learning algorithms on the software vulnerability data by using PPDaaS (Prediction Platform as a Service) show the efficacy of various classifiers. Each of the classification methods tested with a different

feature selection method as shown in Table 4. The performance evaluation of these models is based on F1 Score, Accuracy, and Precision.

5.2.1 Function Code Level

Firstly, for the whole dataset where year range 2002-2019, the Random Forest Classifier gives the best results with feature selection models with SelectKBest, PCA, and Recursive Feature Elimination with Cross-Validation (RFECV). The top accuracy is nearly 95.08 and precision recall and F1 score is nearly close to 0.94.

Between the years 2002 to 2014, the Stacking Classifier and Random Forest Classifier showed commendable performance achieving high results. Accuracy hovered around 0.95 and Precision hovered around 0.939. By that time Bernoulli NB Passive Aggressive Classifier gave the worst performance at around 0.73 and 0.55 respectively.

The best performance for that year, optimizing model performance for this data is the Stacking Classifier with and Sparse PCA Feature Selection Method and the second is the Random Forest Classifier with Mini Batch Sparse PCA.

In the period between 2002 to 2016, the Random Forest Classifier algorithm once again gave the best results. However, this time it was accompanied by different feature selection models, such as Recursive Feature Elimination (RFE) and SelectKBest. The highest model performance is taken with evolution metrics with an accuracy of about 95.71% and the prediction, recall and F1 score is nearly the same around 0.95.

The authors Pereira and colleagues [9] applied several classification methods to the Software Vulnerability dataset. They used several classifiers and hyperparameter settings, they provided their best precision, recall, and F1-score results. They obtained the best Precision score with the Random Forest algorithm using the data balancing technique in addition to this, they obtained the best Recall score on the Bagging algorithm, and with not using the data balancing technique and dimension reduction used variance. Lastly, they took the best F1-score with the Random Forest algorithm without using the Data balancing technique and using dimension reduction as Variance. Their best Precision is 0.9362, their best Recall is 0.9505, and best F1-score is 0.9379.

Table 33: Comparison between our results versus Pereira and colleague's results

Platform	Precision	Recall	F1 Score
Pereira and colleagues	0.9362	0.9505	0.9379
PPDaaS	0.951828	0.951068	0.93886

Our outcomes were superior to their highest precision, recall, and F1-score performances. The SGD Classifier model in our analysis had the highest precision with 0.951828. Similarly, the Stacking Classifier outperformed the Bagging No Variance bootstrap model from the last trial, achieving our best recall with a value of 0.951068. Our top-performing model in terms of the F1-score measure was the Random Forest Classifier, which outperformed their RF No Variance bootstrap modal with an F1-score of 0.93886.

We used various feature selection methods to select the best method for the data, in that way we observed best feature selection methods are for the evaluation metrics SelectKBest with mutual_info_classif for best F1-Score, Sparse PCA for best Recall, RFECV with an estimator Random Forest Classifier and hyperparameter tuning as number of features set to 28 for best Precision and Sparse PCA for the best accuracy.

We used the model degradation feature (Fig 15) to analyze which model is degraded or improved in a specific period and evaluated them.

Model	Feature Selection File 1	Feature Selection File 1	Metric	File 1 Value	File 2 Value	Difference	Status
RandomForestClassifier	PCA(n_components=28)	PCA(n_components=28)	Accuracy	0.9383355931	0.9475297473	0.009194154199999915	Positive
RandomForestClassifier	RFE(estimator=MLPClassifier(), n_features_to_select=28)	RFE(estimator=MLPClassifier(), n_features_to_select=28)	Accuracy	0.9350577625	0.9474503525	0.012392590000000037	Positive
RandomForestClassifier	SelectKBest(mutual_info_classif, k=28)	SelectKBest(mutual_info_classif, k=28)	Accuracy	0.9357382239	0.9473301687	0.01159194479999992	Positive
RandomForestClassifier	RFE(estimator=SVC(), n_features_to_select=28)	RFE(estimator=SVC(), n_features_to_select=28)	Accuracy	0.935864876	0.9473301687	0.011465292699999963	Positive
StackingClassifier	RFE(estimator=SVC(), n_features_to_select=28)	RFE(estimator=SVC(), n_features_to_select=28)	Accuracy	0.9356931526	0.9471390524	0.011445899799999992	Positive
StackingClassifier	RFECV(estimator=SVC(), min_features_to_select=28)	RFECV(estimator=SVC(), min_features_to_select=28)	Accuracy	0.9363537628	0.9469010989	0.010547336099999916	Positive
StackingClassifier	MiniBatchSparsePCA(n_components=28)	MiniBatchSparsePCA(n_components=28)	Accuracy	0.9381173256	0.9468807932	0.008763467599999908	Positive
RandomForestClassifier	RFECV(estimator=LogisticRegression(), min_features_to_select=28)	RFECV(estimator=LogisticRegression(), min_features_to_select=28)	Accuracy	0.9359925648	0.9468799813	0.010887416499999913	Positive
StackingClassifier	SelectKBest(f_regression, k=28)	SelectKBest(f_regression, k=28)	Accuracy	0.9359744893	0.9468137219	0.01083923259999997	Positive
RandomForestClassifier	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)	Accuracy	0.9347404046	0.9468018279	0.01206142330000004	Positive
RandomForestClassifier	SelectKBest(f_regression, k=28)	SelectKBest(f_regression, k=28)	Accuracy	0.9362250955	0.9466663851	0.010441289600000014	Positive
RandomForestClassifier	RFE(estimator=LogisticRegression(), n_features_to_select=28)	RFE(estimator=LogisticRegression(), n_features_to_select=28)	Accuracy	0.9349913433	0.9466192342	0.011627890900000049	Positive
StackingClassifier	RFE(estimator=MLPClassifier(), n_features_to_select=28)	RFE(estimator=MLPClassifier(), n_features_to_select=28)	Accuracy	0.9362615468	0.9465542113	0.010292664499999993	Positive
StackingClassifier	SparsePCA(n_components=28)	SparsePCA(n_components=28)	Accuracy	0.9384780947	0.9465542113	0.00807611659999996	Positive
RandomForestClassifier	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)	Accuracy	0.9349920969	0.9464880872	0.011495990300000036	Positive
StackingClassifier	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)	Accuracy	0.9363109095	0.9464536977	0.01014278820000003	Positive
StackingClassifier	RFECV(estimator=LogisticRegression(), min_features_to_select=28)	RFECV(estimator=LogisticRegression(), min_features_to_select=28)	Accuracy	0.9368274107	0.9464376284	0.009610217700000034	Positive
StackingClassifier	RFE(estimator=LogisticRegression(), n_features_to_select=28)	RFE(estimator=LogisticRegression(), n_features_to_select=28)	Accuracy	0.936164601	0.9463039991	0.010139398099999974	Positive

Figure 15: Model Degradation Tool Output

Table 34 shows that the Passive Aggressive Classifier takes the lead for model improvement with a value of 0.047 between 2002-2014 to 2002-2016 period.

Table 34: 2002-2014 to 2002-2016 Model Degradation Top 4 Model Improvement Results

Model	**Feature Selection**	**Metric**	**2002-2014**	**2002-2016**	**Difference**	**Status**
Passive Aggressive Classifier	RFE estimator Random Forest Classifier	Accuracy	0.869909508	0.917359817 4	0.047450 3094	Negative
Passive Aggressive Classifier	Mini Batch Sparse PCA	Accuracy	0.881254073	0.919252408 7	0.037998 3357	Negative
Passive Aggressive Classifier	RFECV estimator SVC	Accuracy	0.901151606 7	0.921118425 3	0.019966 8196	Negative
Passive Aggressive Classifier	RFE estimator SVC	Accuracy	0.907637684 1	0.924677541	0.017039 8569	Negative

Table 35: 2002-2014 to 2002-2016 Model Degradation Top 4 Model Deterioration Results

Model	**Feature Selection**	**Metric**	**2002-2014**	**2002-2016**	**Difference**	**Status**
GaussianNB	SelectKBest with f_classif	Accuracy	0.915628	0.241069	-0.67456	Positive
GaussianNB	SelectKBest with mutual_info_classif	Accuracy	0.915628	0.241069	-0.67456	Positive
GaussianNB	SelectKBest with f_regression	Accuracy	0.915628	0.241069	-0.67456	Positive
GaussianNB	RFE with estimator Decision Tree Classifier	Accuracy	0.915628	0.241069	-0.67456	Positive

Passive Aggressive Classifier with RFE using Random Forest Classifier is an estimator and selecting 28 features shows us the Accuracy in 2002-2014 as 0.869909508. On the other hand, Accuracy in 2002-2016 is 0.9173598174. The difference is 0.0474503094 which means the accuracy of the model has shown a

significant improvement of 0.0474503094 between in 2016. This suggests that the RFE feature selection technique, coupled with the Random Forest Classifier as the estimator, has positively improved over time. On the other hand, the GaussianNB model seems to degrade significantly in performance (in terms of accuracy) when trained on data from the period 2002-2014 compared to the period 2002-2016 shown in Table 35.

Passive Aggressive Classifier with SelectKBest feature selection using f_classif scoring and selecting 28 best features shows us the accuracy in 2002-2016 as 0.8765474036. On the other hand, accuracy in 2002-2019 is 0.917257212. The difference is 0.0407098084 which means the model's performance has improved by 0.0407098084 between 2002 and 2019, indicating the added data improved the model prediction capability. Table 36 shows the top four model improvement results compared with the data from 2002-2016 to 2002-2019. On the other hand, The GaussianNB model, when evaluated using different feature selection techniques, shows a consistent degradation in performance when transitioning from data of the period 2002-2016 to 2002-2019 shown in Table 37.

Table 36: 2002-2016 to 2002-2019 Model Degradation Top 4 Model Improvement Results

Model	**Feature Selection**	**Metric**	**2002-2016**	**2002-2019**	**Difference**	**Status**
Passive Aggressive Classifier	SelectKBest with f_classif	Accuracy	0.8765474036	0.917257212	0.0407098084	Negative
SGD Classifier	RFECV with estimator Random Forest Classifier	Accuracy	0.9285697598	0.9518282808	0.023258521	Negative
SGD Classifier	RFECV with estimator Decision Tree Classifier	Accuracy	0.9061690468	0.9179257369	0.0117566901	Negative

Ada Boost Classifier	Mini Batch Sparse PCA	Accuracy	0.9218097851	0.926350702	0.0045409169	Negative

Table 37: 2002-2016 to 2002-2019 Model Degradation Top 4 Model Deterioration Results

Model	**Feature Selection**	**Metric**	**2002-2016**	**2002-2019**	**Difference**	**Status**
GaussianNB	SelectKBest with f_classif	Accuracy	0.92492	0.251167	-0.67375	Positive
GaussianNB	SelectKBest with mutual_info_classif	Accuracy	0.92492	0.251167	-0.67375	Positive
GaussianNB	SelectKBest with f_regression	Accuracy	0.92492	0.251167	-0.67375	Positive
GaussianNB	RFE with estimator Decision Tree Classifier	Accuracy	0.92492	0.251167	-0.67375	Positive

In summary, the results from the analyzed models and feature selection techniques in the years 2002-2014 to 2002-2016 and 2002-2016 to 2002-2019 periods indicate a generally positive trend in model performance over time. These findings are valuable in the context of model maintenance and deployment, as they highlight the importance of choosing appropriate feature selection methods to ensure model robustness over extended periods. On the other hand, the GaussianNB model exhibits a notable decrease in accuracy when expanding the data timeframe, irrespective of the feature selection method used.

5.2.2 Class Code Level

The most significant observation is the consistency of Ridge Classifier CV around all metrics at class code level granularity. We obtained the highest accuracy, precision, recall, and F1 score from it. Moreover, these evaluation metrics are nearly close to each other around 94%. This shows us the class-level dataset tends to be consistent with Ridge Classifier CV and this classifier is robust and reliable for this dataset. In addition to these, we obtained a Random Forest Classifier and Bagging Classifier gave promising results. Although Ridge Classifier CV, on the other hand, performs extraordinarily well in terms of accuracy, precision, recall, and F1 score, the

Random Forest Classifier had the greatest precision score, which was around 93.69%. With the comparison of best case to worst case, SGDClassifier shows the worst results in every metric. This shows that Ridge Classifier CV was capable of accurately detecting vulnerabilities and SGDClassifier was the worst. The dataset used is gathered from the study [28] for this data granularity. Pereira and colleagues [28] did not perform an analysis of class-level vulnerabilities; however, we performed an analysis of this gap.

5.2.3 File Code Level

The top 5 results of the file-level vulnerability dataset results show that feature selection methods including SelectKBest with f_classif and mutual_info_classif and RFE with Decision Tree Classifier and Random Forest Classifier have no significant effect on accuracy, as the highest accuracy is approximately 95.48%. In addition to that, precision differs from it. Hist Gradient Boosting Classifier achieved the highest precision of approximately 95%. However, precision differs with different feature selection methods, the highest score was achieved with RFECV with estimator Random Forest Classifier. By looking at the Recall, the same results are seen with Accuracy. Lastly, the highest F1 Score is achieved by the Random Forest Classifier, and the Top 5 is the same with the highest result. The top 5 results differ because of the feature selection methods. The best result of the F1 Score was achieved with the feature selection method the RFECV with the estimator SVC. Pereira and colleagues [28] did not perform an analysis of file-level vulnerabilities; however, we performed an analysis of this gap.

Overall, the analysis indicates that the models achieved high accuracy and balanced performance in terms of Precision, Recall, and F1 Score. Different machine learning algorithms were employed with varying feature selection methods, showcasing the potential of these models in detecting vulnerabilities in the software dataset.

In conclusion, the optimal choice of machine learning algorithms and feature selection techniques can significantly vary based on the granularity level and the performance metric in focus as shown in Table 38.

Table 38: Best ML Algorithm and Feature Selection according to Metrics in all Granularity

	Function		Class		File	
Metric	**Algorithm**	**Feature Selection Technique**	**Algorithm**	**Feature Selection Technique**	**Algorithm**	**Feature Selection Technique**
Accuracy	Stacking Classifier	Sparse PCA	Ridge Classifier CV	SelectKBest with f_classif	XGB Classifier	SelectKBest with f_classif
Precision	SGD Classifier	RFECV with estimator Random Forest Classifier	Random Forest Classifier	RFECV with estimator SVC	Hist Gradient Boosting Classifier	RFECV with estimator Random Forest Classifier
Recall	Stacking Classifier	Sparse PCA	Ridge Classifier CV	SelectKBest with f_classif	XGB Classifier	SelectKBest with f_classif
F1 Score	Random Forest Classifier	SelectKBest with mutual_info_classif	Random Forest Classifier	RFECV with estimator SVC	Random Forest Classifier	RFECV with estimator SVC

5.2.4 Usability Analysis

The PPDaaS software platform received predominantly positive feedback from the survey respondents. Most users, across varied experience levels and age groups, were satisfied with its capabilities and found it user-friendly and effective for automated machine learning. The survey questions and answers with an average score are shown in Table 39 which indicates a generally positive perception of the developed platform PPDaaS especially, among those in the age group of 25-44 years with significant experience in the sector. Most participants believe the platform is effective and the average score is 4.3; user-friendly with an average score of 3.8, and offers considerable utility. On average, the platform's cost-saving potential stood out the most with a rating of 4.3. The overall satisfaction level and time-saving potential also received high averages, ranging from 4.0 to 4.5. These averages show that feedback from the user is generally positive.

Table 39: Survey Results with Average Score

Questions	**Average of the Answers**
Overall, how satisfied are you with the PPDaaS software platform? (1=not at all satisfied, 5=very satisfied)	4.4
How easy was it to understand how to use the PPDaaS software platform? (1=very difficult, 5=very easy)	3.8
How user-friendly is the PPDaaS software platform? (1=not at all user-friendly, 5=very user-friendly)	3.8
How effective is the PPDaaS software platform at helping you build an automated machine learning model? (1=not effective at all, 5=very effective)	4.3
How useful is the PPDaaS software platform in terms of its capabilities and features? (1=not at all useful, 5=very useful)	4. 2
How likely are you to recommend the PPDaaS software platform to a colleague or friend? (1=not at all likely, 5=very likely)	4.3
How likely are you to use the PPDaaS software platform in the future for building automated machine learning models? (1=not at all likely, 5=very likely)	4.1
How much did you enjoy using the PPDaaS software platform? (1=not at all enjoyable, 5=very enjoyable)	3.8
How much did you learn about automated machine learning through this video on the PPDaaS software platform? (1=not at all, 5=very much)	3.9
How much will the PPDaaS software platform help you save time and effort in building automated machine learning models? (1=not at all, 5=very much)	4.3
How much will the PPDaaS software platform help you improve the quality of your automated machine learning models? (1=not at all, 5=very much)	4.1
How much will the PPDaaS software platform help you reduce the cost of building automated machine learning models? (1=not at all, 5=very much)	4.7

CHAPTER VI
CONCLUSION

In conclusion, in this thesis, an AutoML platform named PPDaaS is developed that incorporates runtime limit and model degradation features. Initially, we explored and highlighted the absence of features in the available platforms. PPDaaS enables users to apply and use feature selection techniques and machine learning models without writing any code automatically. Users can compare their model performance results around 45 Machine Learning Algorithms (Table 2, Table 3), each run with different feature selection methods (Table 4). Because software vulnerability has a critical role in the software industry [1], we use the software vulnerability data created by [28] to build prediction models using PPDaaS. In addition to this, we conducted tests on the OpenML adult dataset to make comparisons with other automatic machine-learning tools such as Weka[15], and H2O.ai[16].

Firstly, we analyzed the performance of all granularity levels of the vulnerability dataset which are functions, classes, and files. Our study showed that the levels of granularity of the vulnerability dataset performed well nearly the same in predicting the vulnerability. Especially, class-based level has the highest performance in classifying the vulnerability based on the F1 Score. In addition to that, machine learning algorithms also showed differences in performance for each granularity.

Furthermore, an experiment is conducted on OpenML adult data to make a benchmark analysis and compare it with the other AutoML tools. The results show that the PPDaaS platform evaluation value as 1-ROC is very close to the platform that was created by [14]. However, we evaluate the model performance with different metrics such as Recall, Precision, and F1-Score. The model has the best performance in the SDGClassifier and Passive Aggressive Classifier.

Afterward, because the vulnerability data may change over time, we simulated model degradation analysis on the function level data by splitting it into three chunks (2002 to 2014, 2002 to 201, and 2002 to 2019) and developed models for those datasets. The comprehensive analysis of the model’s performance for the periods

2002-2014 to 2002-2016 and 2002-2016 to 2002-2019 reveal a consistent trend of improved in model performance over time. Our results revealed that Aggressive Classifiers have a positive trend in model performance evaluation over time. On the other hand, the GaussianNB model demonstrates a comparable and significant degradation in performance when transitioning from the earlier dataset to the later one. This consistency in performance drop is observed irrespective of the feature selection method applied. The observed positive outcomes have a vital role in selecting suitable feature selection methods and machine learning algorithms to protect resilience and accuracy over an extended duration.

Overall, we have successfully addressed the research questions by providing new solutions for the "Runtime Limit" and Model Degradation features for the PPDaaS platform that fills the gap in the AutoML literature. When applied to software vulnerability data, our experiment results outperform Pereira and colleagues [28] results for software vulnerability prediction. Furthermore, our data indicates that software vulnerability prediction is optimal at the class granularity level. Additionally, the model performance has displayed a generally positive trend over the specific periods examined.

Lastly, to get feedback about the capabilities of this platform we conducted a study to gather users' views on the platform. The platform is evaluated by out of 10 software professionals. Overall, their answers highlight that our platform can find a place in the software industry because of its capabilities ease of use, and applicability to real-world scenarios.

6.1 FUTURE WORK

In the next phase of this research, the AutoML prediction results can be visually enhanced by introducing complex visualization techniques. Not only will these visual aids help specialists in the field to quickly understand the results, but they can also make the research more comprehensible for non-experts. Additionally, automatic interpretations of the machine learning results could be added to the tool.

REFERENCES

[1] Jordan M. I. and Mitchell T. M. (2015), "Machine learning: Trends, perspectives, and prospects", *Science*, vol. 349, no. 6245, pp. 253–255, doi: 10.1126/science.aac4520.

[2] Gong Z., Zhong P., and Hu W. (2019), "Diversity in Machine Learning", *IEEE Access*, vol. 7, pp. 64323–64350, doi: 10.1109/ACCESS.2019.2917620.

[3] Lee Y. W., Choi J. W., and Shin E. H. (2021), "Machine learning model for predicting malaria using clinical information", *Comput Biol Med*, vol. 129, pp. 1-7, doi: 10.1016/j.compbiomed.2020.104151.

[4] Arfat Y., Mittone G., Esposito R., Cantalupo B., Deferrari G. M., and Aldinucci M. (2022), "Machine learning for cardiology", *Minerva Cardiology and Angiology*, vol. 70, no. 1, pp. 75–91, doi: 10.23736/S2724-5683.21.05709-4.

[5] Greener J. G., Kandathil S. M., Moffat L., and Jones D. T. (2022), "A guide to machine learning for biologists", *Nature Reviews Molecular Cell Biology*, vol. 23, no. 1, pp. 40–55, doi: 10.1038/s41580-021-00407-0.

[6] Goodell J. W., Kumar S., Lim W. M., and Pattnaik D. (2021), "Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis", *Journal of Behavioral and Experimental Finance*, vol. 32, pp. 1-19, doi: 10.1016/j.jbef.2021.100577.

[7] Rundo F., Trenta F., di Stallo A. L., and Battiato S. (2019), "Machine learning for quantitative finance applications: A survey", *Applied Sciences (Switzerland)*, vol. 9, no. 24, pp. 1-20, doi: 10.3390/app9245574.

[8] Gogas P. and Papadimitriou T. (2021), "Machine Learning in Economics and Finance", *Computational Economics*, vol. 57, no. 1, pp. 1-4, doi: 10.1007/s10614-021-10094-w.

[9] Anute N., Paliwal M., Patel M., and Kandale N. (2021), “Impact of artificial intelligence and machine learning on business operations”, *Journal of Management Research and Analysis*, vol. 8, no. 2, pp. 69–74, doi: 10.18231/j.jmra.2021.015.

[10] Sarker I. H. (2021), “Machine Learning: Algorithms, Real-World Applications and Research Directions”, *SN Computer Science*, vol. 2, no. 3, pp. 1-21, doi: 10.1007/s42979-021-00592-x.

[11] Lebens M., Finnegan R., Sorsen S., and Shah J. (2021), “Rise of the citizen developer”, *Muma Business Review*, vol. 5, no. 12, pp. 101-111, doi:10.28945/4885.

[12] Hanussek M., Blohm M., and Kintz M. (2021), “Can AutoML outperform humans? An evaluation on popular OpenML datasets using AutoML Benchmark”, *In 2020 2nd International Conference on Artificial Intelligence, Robotics and Control*, pp. 29–32, New York, USA, doi: 10.1145/3448326.3448353.

[13] Musigmann M., Akkurt B.H., Krähling H., Nacul N.G., Remonda L., Sartoretti T., Henssen D., Brokinkel B., Stummer W., Heindel W., and Mannil M. (2022), “Testing the applicability and performance of Auto ML for potential applications in diagnostic neuroradiology”, *Sci Rep*, vol. 12, no. 1, pp. 1-21, doi: 10.1038/s41598-022-18028-8.

[14] Feurer M., Eggensperger K., Falkner S., Lindauer M., and Hutter F. (2022), “Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning”, *Journal of Machine Learning Research*, vol. 23, pp. 1-61, doi: 10.48550/arXiv.2007.04074.

[15] Witten I. H., Frank E., Trigg L. E., Hall M. A., Holmes G., and Cunningham S. J. (1999), “Weka: Practical machine learning tools and techniques with Java implementations”, *Computer Science Working Papers*, pp. 1-5.

[16] Cook D. (2016), *Practical Machine Learning with H2O*, 1st Edition, O'Reilly Media Inc., Sebastopol.

[17] Krishnan S., Franklin M. J., Goldberg K., and Wu E. (2017), “BoostClean: Automated Error Detection and Repair for Machine Learning”, *arXiv*, pp. 1-15, doi: 10.48550/arXiv.1711.01299.

[18] Krummenacher G., Ong C. S., Koller S., Kobayashi S., and Buhmann J. M. (2018), “Wheel Defect Detection with Machine Learning”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1176–1187, doi: 10.1109/TITS.2017.2720721.

[19] Harer J. A., Kim L. Y., Russell R. L., Ozdemir O., Kosta L. R., Rangamani A., Hamilton L. H., Centeno G. I., Key J. R., Ellingwood P. M., McConley M. W., Opper J. M., Chin P., and Lazovich T. (2018). "Automated software vulnerability detection with machine learning", *arXiv*, pp. 1-8, doi: 10.48550/arXiv.1803.04497.

[20] Medeiros N., Ivaki N., Costa P., and Vieira M. (2020), “Vulnerable Code Detection Using Software Metrics and Machine Learning”, *IEEE Access*, vol. 8, pp. 219174–219198, doi: 10.1109/ACCESS.2020.3041181.

[21] Eberendu A. C., Udegbe V. I., Ezennorom E. O., Ibegbulam A. C., and Chinebu T. I. (2022), “A Systematic Literature Review of Software Vulnerability Detection”, *European Journal of Computer Science and Information Technology*, vol. 10, no. 1, pp. 23-37.

[22] Shar L. K., Briand L. C., and Tan H. B. K. (2015), “Web Application Vulnerability Prediction Using Hybrid Program Analysis and Machine Learning”, *IEEE Trans Dependable Secure Comput*, vol. 12, no. 6, pp. 688–707, doi: 10.1109/TDSC.2014.2373377.

[23] Hydara I., Sultan A. B. M., Zulzalil H., and Admodisastro N. (2015), “Current state of research on cross-site scripting (XSS) - A systematic literature review”, *Information and Software Technology*, vol. 58, pp. 170–186, doi: 10.1016/j.infsof.2014.07.010.

[24] Zeng P., Lin G., Pan L., Tai Y., and Zhang J. (2020), “Software vulnerability analysis and discovery using deep learning techniques: A survey”, *IEEE Access*, vol. 8, pp. 197158–197172, doi: 10.1109/ACCESS.2020.3034766.

[25] Shah I. A., Rajper S., and ZamanJhanjhi N. (2021), “Using ML and Data-Mining Techniques in Automatic Vulnerability Software Discovery”, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, no. 3, pp. 2109–2126, doi: 10.30534/ijatcse/2021/871032021.

[26] Dowd M., McDonald J., and Schuh J. (2006), *The Art of Software Security Assessment-Identifying and Preventing Software Vulnerabilities*, 1st Edition, O'Reilly Media Inc., Ann Arbor.

[27] Bagnato A. (2009), "Security in Model-Driven Architecture", *CTIT Workshop Proceedings Series*, pp. 5-42, Enschede, Netherlands.

[28] Pereira J. D., Antunes J. H., and Vieira M. (2022), "A Software Vulnerability Dataset of Large Open Source C/C++ Projects", *Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing*, pp. 152–163, Beijing, China, doi: 10.1109/PRDC55274.2022.00029.

[29] Sultana K. Z., Anu V., and Chong T. Y. (2021), "Using software metrics for predicting vulnerable classes and methods in Java projects: A machine learning approach", *Journal of Software: Evolution and Process*, vol. 33, no. 3, pp. 1-20, doi: 10.1002/smr.2303.

[30] Vanschoren J., van Rijn J. N., Bischl B., and Torgo L. (2013), "OpenML: networked science in machine learning", *SIGKDD Explor. Newsl.*, vol. 15, no. 2, pp. 49–60, doi: 10.1145/2641190.2641198.

[31] Pereira J. D. A. (2020), "Techniques and Tools for Advanced Software Vulnerability Detection", *Proceedings - 2020 IEEE 31st International Symposium on Software Reliability Engineering Workshops, ISSREW 2020*, pp. 123–126, Coimbra, Portugal, doi: 10.1109/ISSREW51248.2020.00049.

[32] Pereira J. D. A., Lourenço N., and Vieira M. (2023), "On the Use of Deep Graph CNN to Detect Vulnerable C Functions", *ACM International Conference Proceeding Series*, pp. 45–50, New York, USA, doi: 10.1145/3569902.3569913.

[33] Elkhail A. A. and Cerny T. (2019), "On Relating Code Smells to Security Vulnerabilities", *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity)*, pp. 7–12, Washington, USA, doi: 10.1109/BigDataSecurity-HPSC-IDS.2019.00013.

[34] Alves H., Fonseca B., and Antunes N. (2016), "Software Metrics and Security Vulnerabilities: Dataset and Exploratory Study", *Proceedings - 2016 12th European Dependable Computing Conference, EDCC 2016*, pp. 37–44, Gothenburg, Sweden, doi: 10.1109/EDCC.2016.34.

[35] Medeiros N., Ivaki N., Costa P., and Vieira M. (2021), “An Empirical Study on Software Metrics and Machine Learning to Identify Untrustworthy Code”, *Proceedings - 2021 17th European Dependable Computing Conference, EDCC 2021*, pp. 87–94, Munich, Germany, doi: 10.1109/EDCC53658.2021.00020.

[36] Zhou A., Sultana K. Z., and Samanthula B. K. (2021), “Investigating the Changes in Software Metrics after Vulnerability is Fixed”, *Proceedings - 2021 IEEE International Conference on Big Data, Big Data 2021*, pp. 5658–5663, Orlando, USA, doi: 10.1109/BigData52589.2021.9671334.

[37] Siavvas M. (2019), *Static Analysis for Facilitating Secure and Reliable Software* (Doctoral Dissertation), Imperial College London, London.

[38] Kaya A., Keceli A. S., Catal C., and Tekinerdogan B. (2019). “The impact of feature types, classifiers, and data balancing techniques on software vulnerability prediction models”, *J. Softw. Evol. Process*, pp. 1–25, doi: 10.1002/smr.2164.

[39] Sultana K.Z., Boyd C.B., and Williams B.J. (2023), "A Software Vulnerability Prediction Model Using Traceable Code Patterns and Software Metrics", *SN COMPUT. SCI.*, vol. 4, pp. 1-23, doi: 10.1007/s42979-023-02077-5.

[40] Gupta A., Suri B., Kumar V., and Jain P. (2021), “Extracting rules for vulnerabilities detection with static metrics using machine learning”, *International Journal of System Assurance Engineering and Management*, vol. 12, no. 1, pp. 65–76, doi: 10.1007/s13198-020-01036-0.

[41] S. Lipp, S. Banescu, and A. Pretschner (2022), “An Empirical Study on the Effectiveness of Static C Code Analyzers for Vulnerability Detection”, *ISSTA 22*, vol. 12, pp. 1-12, doi: 10.1145/3533767.

[42] Ban X., Liu S., Chen C., and Chua C. (2019), “A performance evaluation of deep-learnt features for software vulnerability detection”, *Concurrency and Computation: Practice and Experience*, vol. 31, no. 19, pp. 1-10, doi: 10.1002/cpe.5103.

[43] Shukla A., Katt B., and Nweke L.O. (2019), "Vulnerability Discovery Modelling With Vulnerability Severity", *2019 IEEE Conference on Information and Communication Technology*, pp. 1-6, Allahabad, India, doi: 10.1109/CICT48419.2019.9066187.

[44] Bassi D. and Singh H. (2023), “The Effect of Dual Hyperparameter Optimization on Software Vulnerability Prediction Models”, *E-Informatica Software Engineering Journal*, vol. 17, no. 1, pp. 230102, doi: 10.37190/e-Inf230102.

[45] Pereira J.D., Antunes J.H., and Vieira M. (2021), "On Building a Vulnerability Dataset with Static Information from the Source Code", *2021 10th Latin-American Symposium on Dependable Computing (LADC)*, pp. 1-2, Florianópolis, Brazil, doi: 10.1109/LADC53747.2021.9672589.

[46] Ugrenovic D., Vankeirsbilck J., Pissoort D., Holvoet T., and Boydens J. (2020), "Designing Out-of-distribution Data Detection using Anomaly Detectors: Single Model vs. Ensemble", *2020 XXIX International Scientific Conference Electronics (ET)*, pp. 1-4, Sozopol, Bulgaria, doi: 10.1109/ET50336.2020.9238227.

[47] Ugrenovic D. et al. (2021), "Towards classification trustworthiness: one-class classifier ensemble", *2021 XXX International Scientific Conference Electronics (ET)*, pp. 1-6, Sozopol, Bulgaria, doi: 10.1109/ET52713.2021.9580016.

[48] Alves H., Fonseca B., and Antunes N. (2016), “Software Metrics and Security Vulnerabilities: Dataset and Exploratory Study”, *Proceedings - 2016 12th European Dependable Computing Conference, EDCC 2016*, pp. 37–44, Gothenburg, Sweden, doi: 10.1109/EDCC.2016.34.

[49] Alves H., Fonseca B., and Antunes N. (2016), “Experimenting machine learning techniques to predict vulnerabilities”, *Proceedings - 7th Latin-American Symposium on Dependable Computing, LADC 2016*, pp. 151–156, Cali, Colombia, doi: 10.1109/LADC.2016.32.

[50] Bhandari G., Naseer A., and Moonen L. (2021), “CVEfixes: Automated collection of vulnerabilities and their fixes from open-source software”, *PROMISE 2021 - Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering, co-located with ESEC/FSE 2021*, pp. 30–39, Athens, Greece, doi: 10.1145/3475960.3475985.

[51] Clemente C. J., Jaafar F., and Malik Y. (2018), “Is predicting software security bugs using deep learning better than the traditional machine learning algorithms?”, *Proceedings - 2018 IEEE 18th International Conference on Software Quality, Reliability, and Security, QRS 2018*, pp. 95–102, Lisbon, Portugal, doi: 10.1109/QRS.2018.00023.

[52] Salimi S., Ebrahimzadeh M., and Kharrazi M. (2020), “Improving real-world vulnerability characterization with vulnerable slices”, *PROMISE 2020 - Proceedings of the 16th ACM International Conference on Predictive Models and Data Analytics in Software Engineering, Co-located with ESEC/FSE 2020*, pp. 11–20, New York, USA, doi: 10.1145/3416508.3417120.

[53] Siavvas M., Kehagias Di., and Tzovaras Di. (2018), “A Preliminary Study on the Relationship among Software Metrics and Specific Vulnerability Types”, *Proceedings - 2017 International Conference on Computational Science and Computational Intelligence, CSCI 2017*, pp. 916–921, Las Vegas, USA, doi: 10.1109/CSCI.2017.159.

[54] Campos J.R., Vieira M., and Costa E. (2019), "Propheticus: Machine Learning Framework for the Development of Predictive Models for Reliable and Secure Software", *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 173-182, Berlin, Germany, doi: 10.1109/ISSRE.2019.00026.

[55] Kaya A., Keceli A.S., Catal C., and Tekinerdogan B. (2019), "The impact of feature types, classifiers, and data balancing techniques on software vulnerability prediction models", *Journal of Software: Evolution and Process*, vol. 31, no. 9, pp. 1-25, doi: 10.1002/smr.2164.

[56] Brown T., Mann B., Ryder N., Subbiah M., Kaplan J.D., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., Agarwal S., Herbert-Voss A., Krueger G., Henighan T., Child R., Ramesh A., Ziegler D., Wu J., Winter C., Hesse C., Chen M., Sigler E., Litwin M., Gray S., Chess B., Clark J., Berner C., McCandlish S., Radford A., Sutskever I., and Amodei D. (2020), "Language Models are Few-Shot Learners", *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877-1901.

[57] Xu C., Guo D., Duan N., and McAuley J. (2023), "Baize: An Open-Source Chat Model with Parameter-Efficient Tuning on Self-Chat Data", *arXiv*, pp. 1-10, doi:10.48550/arXiv:2304.01196.

[58] huggingface (2023), *Chat with Baize - a hugging face space by Project-Baize*, https://huggingface.co/spaces/project-baize/chat-with-baize, DoA. 24.8.2023.

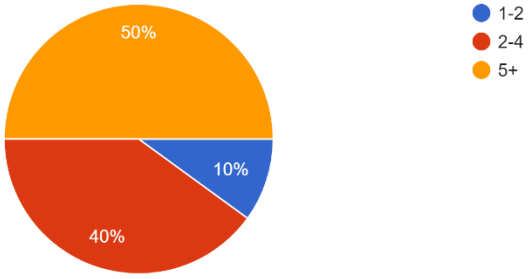
[59] Thapa S. (2023), "Adult Income Prediction Using various ML Algorithms", *SSRN*, vol. 1, no. 1, pp. 1-4, doi: dx.doi.org/10.2139/ssrn.4325813.

APPENDICES

APPENDIX 1: EVALUATION SURVEY RESULTS

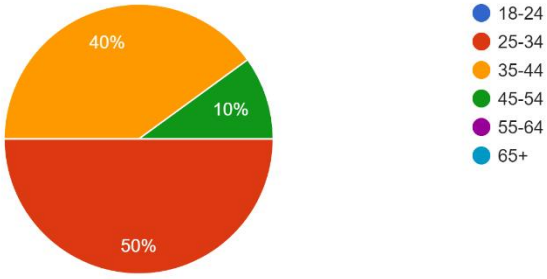
Can you please provide your years of experience in the industry?

10 yanıt



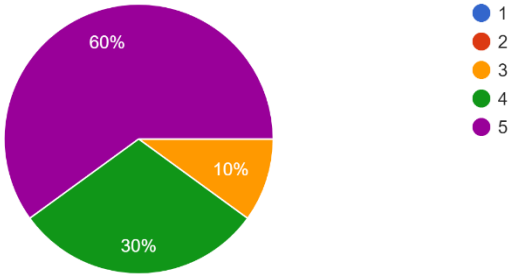
How old are you?

10 yanıt



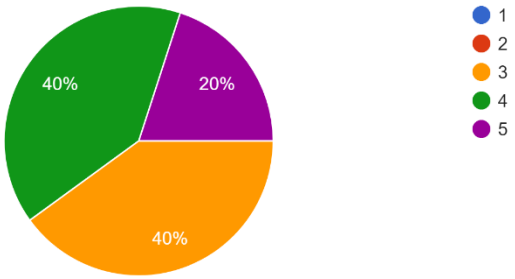
Overall, how satisfied are you with the PPDaaS software platform? (1=not at all satisfied, 5=very satisfied)

10 yanıt



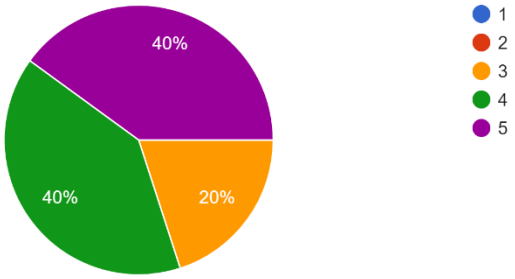
How easy was it to understand how to use the PPDaaS software platform? (1=very difficult, 5=very easy)

10 yanıt



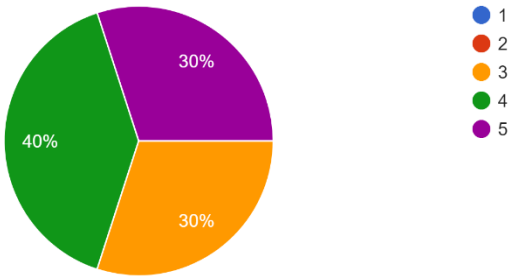
How effective is the PPDaaS software platform at helping you build an automated machine learning model? (1=not effective at all, 5=very effective)

10 yanıt



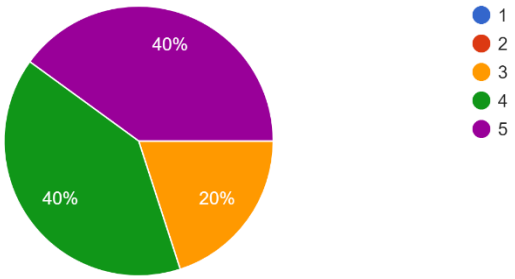
How user-friendly is the PPDaaS software platform? (1=not at all user-friendly, 5=very user-friendly)

10 yanıt



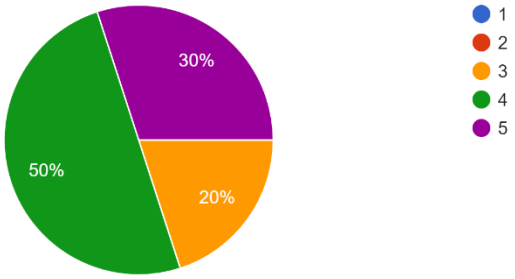
How effective is the PPDaaS software platform at helping you build an automated machine learning model? (1=not effective at all, 5=very effective)

10 yanıt



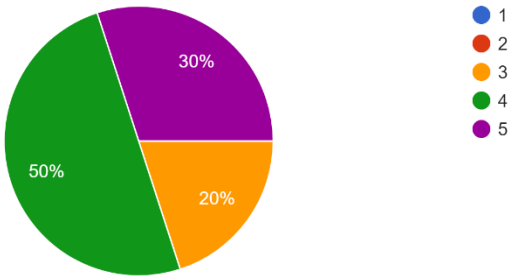
How useful is the PPDaaS software platform in terms of its capabilities and features? (1=not at all useful, 5=very useful)

10 yanıt



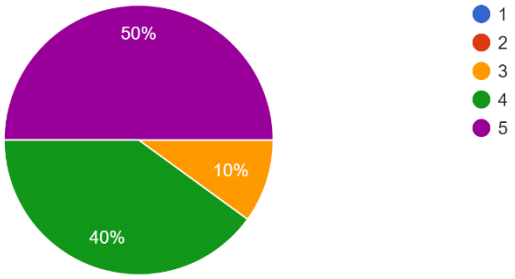
How useful is the PPDaaS software platform in terms of its capabilities and features? (1=not at all useful, 5=very useful)

10 yanıt



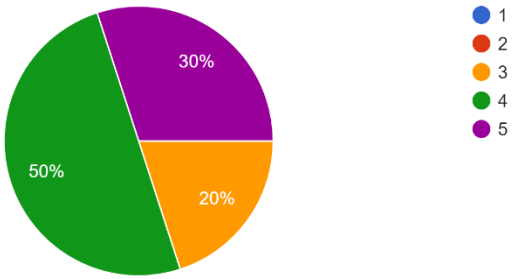
How likely are you to recommend the PPDaaS software platform to a colleague or friend? (1=not at all likely, 5=very likely)

10 yanıt



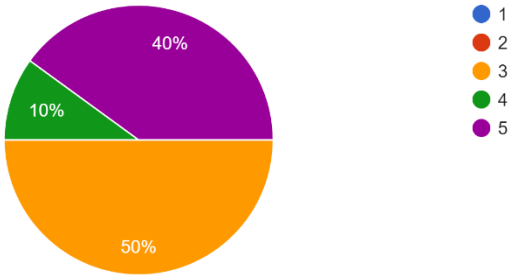
How likely are you to use the PPDaaS software platform in the future for building automated machine learning models? (1=not at all likely, 5=very likely)

10 yanıt



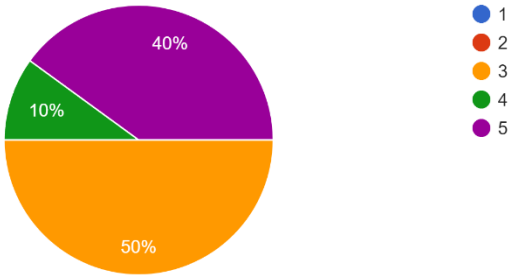
How much did you enjoy using the PPDaaS software platform? (1=not at all enjoyable, 5=very enjoyable)

10 yanıt



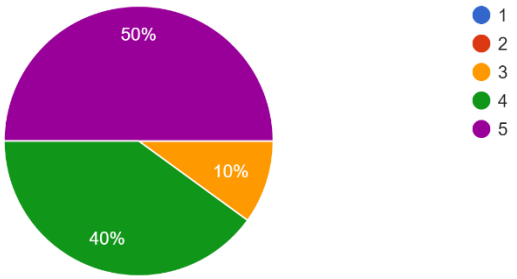
How much did you enjoy using the PPDaaS software platform? (1=not at all enjoyable, 5=very enjoyable)

10 yanıt



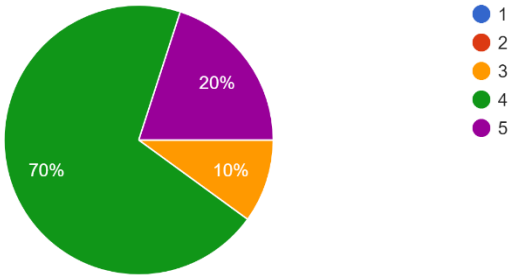
How much will the PPDaaS software platform help you save time and effort in building automated machine learning models? (1=not at all, 5=very much)

10 yanıt



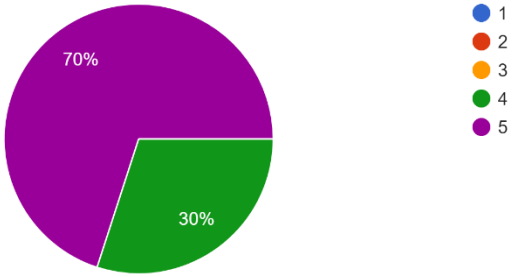
How much will the PPDaaS software platform help you improve the quality of your automated machine learning models? (1=not at all, 5=very much)

10 yanıt



How much will the PPDaaS software platform help you reduce the cost of building automated machine learning models? (1=not at all, 5=very much)

10 yanıt



APPENDIX 2: VULNERABILITY EXPERIMENT FEATURES AND DROP COLUMNS

DROP COLUMNS OF THE VULNERABILITY EXPERIMENT

Description, COMMIT_HASH, COMMIT_DATE, COMMIT_YEAR, VULNERABILITY_CVE, VULNERABILITY_YEAR, VULNERABILITY_CWE, VULNERABILITY_CATEGORY, ID_Function, P_ID, Complement, BeginLine, EndLine, NameMethod, ID_Class, ID_File, FilePath, Patched, Occurrence, R_ID, Affected, binary_label, grouped_multiclass_label, and Visibility.

ALL FEATURES OF THE VULNERABILITY EXPERIMENT

Description, COMMIT_HASH, COMMIT_DATE, COMMIT_YEAR, VULNERABILITY_CVE, VULNERABILITY_YEAR, VULNERABILITY_CWE, VULNERABILITY_CATEGORY, ID_Function, P_ID, Visibility, Complement, BeginLine, EndLine, NameMethod, ID_Class, ID_File, FilePath, Patched, Occurrence, R_ID, Affected, AltCountLineBlank, AltCountLineCode, AltCountLineComment, CountInput, CountLine, CountLineBlank, CountLineCode, CountLineCodeDecl, CountLineCodeExe, CountLineComment, CountLineInactive, CountLinePreprocessor, CountOutput, CountPath, CountSemicolon, CountStmt, CountStmtDecl, CountStmtEmpty, CountStmtExe, Cyclomatic, CyclomaticModified, CyclomaticStrict, Essential, Knots, MaxEssentialKnots, MaxNesting, MinEssentialKnots, RatioCommentToCode, ELIGIBLE_FOR_ALERTS, COMMIT_HAS_ALERTS, TOTAL_ALERTS, Flawfinder_chown, Flawfinder_gets, Flawfinder_printf, Flawfinder_strcpy, Flawfinder_access, Flawfinder_sscanf, Flawfinder_syslog, Flawfinder_crypt, Flawfinder_strcat, Flawfinder_fprintf, Flawfinder_execl, Flawfinder_vfprintf, Flawfinder_sprintf, Flawfinder_execv, Flawfinder_ShellExecute, Flawfinder__sntprintf, Flawfinder__tcscpy, Flawfinder__tcscat, Flawfinder_WinExec, Flawfinder_random, Flawfinder_getenv, Flawfinder_chroot, Flawfinder_srand, Flawfinder_LoadLibrary, Flawfinder_EnterCriticalSection, Flawfinder_InitializeCriticalSection, Flawfinder_getopt, Flawfinder_char, Flawfinder_memcpy, Flawfinder_atoi, Flawfinder_atol, Flawfinder_open, Flawfinder_fopen, Flawfinder_tmpfile, Flawfinder_wcscpy, Flawfinder_TCHAR, Flawfinder_strlen, Flawfinder_read, Flawfinder_strncat, Flawfinder_mismatch, Flawfinder_strncpy, Flawfinder_umask,

Flawfinder_wcslen, Flawfinder_getc, Flawfinder__tcslen, Flawfinder_readlink, Flawfinder_chmod, Flawfinder_vsnprintf, Flawfinder_snprintf, Flawfinder_vsprintf, Flawfinder_vsscanf, Flawfinder_getlogin, Flawfinder_execvp, Flawfinder_execlp, Flawfinder_fscanf, Flawfinder_popen, Flawfinder_vprintf, Flawfinder_system, Flawfinder_wprintf, Flawfinder_scanf, Flawfinder_realpath, Flawfinder_srandom, Flawfinder_getopt_long, Flawfinder_bcopy, Flawfinder_mkstemp, Flawfinder_vfork, Flawfinder_memalign, Flawfinder_usleep, Flawfinder_getchar, Flawfinder_fgetc, Flawfinder_equal, Flawfinder_vfscanf, Flawfinder_setstate, Flawfinder_tmpnam, Flawfinder_wchar_t, Flawfinder_ulimit, Flawfinder_ssignal, Flawfinder_CopyMemory, Flawfinder_wcscat, Flawfinder__snprintf, Flawfinder_vswprintf, Flawfinder_swprintf, Flawfinder__ftprintf, Flawfinder_lstrcpyW, Flawfinder_crypt_r, Flawfinder_lstrcpy, Flawfinder_lstrcpyA, Flawfinder_fwprintf, Flawfinder__stprintf, Flawfinder__vftprintf, Flawfinder_vfwprintf, Flawfinder__vsntprintf, Flawfinder__vstprintf, Flawfinder__tscanf, Flawfinder_wscanf, Flawfinder_SetThreadToken, Flawfinder_lstrcatW, Flawfinder_getwd, Flawfinder_LoadLibraryEx, Flawfinder_GetTempFileName, Flawfinder_CreateProcess, Flawfinder_lrand48, Flawfinder_AddAccessAllowedAce, Flawfinder_drand48, Flawfinder_curl_getenv, Flawfinder_MultiByteToWideChar, Flawfinder__wtoi, Flawfinder__wtoi64, Flawfinder_wcsncpy, Flawfinder_lstrcpyn, Flawfinder__tcsncpy, Flawfinder_wcsncat, Flawfinder_execle, Flawfinder_nrand48, Flawfinder_lstrcpynA, Flawfinder_getpass, Flawfinder_fwscanf, Flawfinder_vwprintf, Flawfinder_CreateProcessAsUser, Flawfinder_g_get_home_dir, Flawfinder_mktemp, Flawfinder_lstrcat, Flawfinder_g_get_tmp_dir, Flawfinder__mbscat, Flawfinder_lstrcatA, Flawfinder_tempnam, multiclass_label, binary_label, grouped_multiclass_label

APPENDIX 3: VULNERABILITY FUNCTION LEVEL DATA RESULTS IN 2002 TO 2014

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
StackingClassifier	0.95043573	0.938478095	0.95043573	0.9399229946	SparsePCA(n_components=28)
RandomForestClassifier	0.950254176	0.938236295	0.950254176	0.9398731333	MiniBatchSparsePCA(n_components=28)
StackingClassifier	0.950254176	0.938117326	0.950254176	0.9396171114	MiniBatchSparsePCA(n_components=28)
RandomForestClassifier	0.95043573	0.938335593	0.95043573	0.939578511	PCA(n_components=28)
StackingClassifier	0.950163399	0.937935685	0.950163399	0.9394638001	PCA(n_components=28)
RandomForestClassifier	0.950072622	0.937713695	0.950072622	0.9392232263	SparsePCA(n_components=28)
StackingClassifier	0.949981845	0.93749022	0.949981845	0.9389812544	SelectKBest (mutual_info_classif, k=28)
StackingClassifier	0.949709513	0.937101168	0.949709513	0.938868844	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
StackingClassifier	0.949618736	0.936827411	0.949618736	0.9385382867	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
BaggingClassifier	0.948438635	0.93561565	0.948438635	0.9384513689	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
RandomForestClassifier	0.94889252	0.935992565	0.94889252	0.9384435089	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
RandomForestClassifier	0.949164851	0.936225096	0.949164851	0.9383825089	SelectKBest (f_regression, k=28)
RandomForestClassifier	0.948801743	0.935864876	0.948801743	0.9383771622	RFE (estimator=SVC(), n_features_to_select=28)
StackingClassifier	0.949255628	0.93631091	0.949255628	0.9383611367	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
RandomForestClassifier	0.948710966	0.935738224	0.948710966	0.9383108611	SelectKBest (mutual_info_classif, k=28)
StackingClassifier	0.949255628	0.936261547	0.949255628	0.938272981	RFE (estimator=MLPClassifier(), n_features_to_select=28)

StackingClassifier	0.94934640 5	0.93635376 3	0.94934640 5	0.93825056 6	RFECV (estimator=SVC(), min_features_to_select=28)
RandomForestClassifier	0.94862018 9	0.93561259 6	0.94862018 9	0.93824460 5	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
BaggingClassifier	0.94798475	0.93510033 5	0.94798475	0.93820345 9	SelectKBest (f_classif, k=28)
StackingClassifier	0.94898329 7	0.93590323 9	0.94898329 7	0.93816244 7	SelectKBest (f_classif, k=28)
StackingClassifier	0.94925562 8	0.93616460 1	0.94925562 8	0.93809531 9	RFE (estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.94852941 2	0.93542705 2	0.94852941 2	0.93809249 2	RFECV (estimator=SVC(), min_features_to_select=28)
StackingClassifier	0.94907407 4	0.93593448 1	0.94907407 4	0.93805213 2	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.94916485 1	0.93597448 9	0.94916485 1	0.93793979 9	SelectKBest (f_regression, k=28)
RandomForestClassifier	0.94834785 8	0.93511588 1	0.94834785 8	0.93787412 4	SelectKBest (f_classif, k=28)
RandomForestClassifier	0.94825708 1	0.93499209 7	0.94825708 1	0.93780817 1	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
StackingClassifier	0.94898329 7	0.93569315 3	0.94898329 7	0.93780791 8	RFE (estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.94816630 4	0.93486930 9	0.94816630 4	0.93774226 2	PCA(n_components=28)
RandomForestClassifier	0.94834785 8	0.93499134 3	0.94834785 8	0.93770042 2	RFE (estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.94843863 5	0.93505776 3	0.94843863 5	0.93767872 2	RFE (estimator=MLPClassifier(), n_features_to_select=28)
RandomForestClassifier	0.94816630 4	0.93474040 5	0.94816630 4	0.93756881 6	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
BaggingClassifier	0.94798475	0.93455997 4	0.94798475	0.93752419 7	RFE (estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.94816630 4	0.93467640 3	0.94816630 4	0.93748144 2	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
RandomForestClassifier	0.94816630 4	0.93461271 1	0.94816630 4	0.93739363 1	RFECV (estimator=MLPClassifier(), min_features_to_select=28)

XGBClassifier	0.949527959	0.936199405	0.949527959	0.937375523	SparsePCA(n_components=28)
BaggingClassifier	0.948075527	0.934486566	0.948075527	0.937328025	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.947530864	0.933965002	0.947530864	0.937196062	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.948983297	0.935347328	0.948983297	0.937169826	PCA(n_components=28)
BaggingClassifier	0.946895425	0.933404378	0.946895425	0.936994080	SelectKBest (mutual_info_classif, k=28)
BaggingClassifier	0.946986202	0.933358301	0.946986202	0.936889395	RFE (estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.946350763	0.933019185	0.946350763	0.936853188	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
BaggingClassifier	0.946532317	0.932897495	0.946532317	0.936648224	SelectKBest (f_regression, k=28)
BaggingClassifier	0.946713871	0.932950951	0.946713871	0.936608101	RFE (estimator=MLPClassifier(), n_features_to_select=28)
BaggingClassifier	0.946441540	0.932463986	0.946441540	0.936240998	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
XGBClassifier	0.948075527	0.933673857	0.948075527	0.936146066	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.946895425	0.932626507	0.946895425	0.936127175	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.946441540	0.932217813	0.946441540	0.935980024	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
BaggingClassifier	0.946350763	0.9310285	0.946350763	0.934739412	SparsePCA(n_components=28)
BaggingClassifier	0.945170661	0.930283629	0.945170661	0.934639454	RFECV (estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	SelectKBest (f_classif, k=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	SelectKBest (mutual_info_classif, k=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	SelectKBest (f_regression, k=28)

XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	RFE (estimator=LogisticRegression(), n_features_to_select=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	RFE (estimator=SVC(), n_features_to_select=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	RFE (estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	RFECV (estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.948166304	0.932159203	0.948166304	0.932592812	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.9347313	0.930163551	0.9347313	0.932321216	MiniBatchSparsePCA(n_components=28)
ExtraTreeClassifier	0.934640523	0.929452045	0.934640523	0.931885393	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.948892520	0.934250874	0.948892520	0.931783184	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.949346405	0.936140963	0.949346405	0.931712582	PCA(n_components=28)
ExtraTreeClassifier	0.934005084	0.929040751	0.934005084	0.931380906	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.948801743	0.934155026	0.948801743	0.931104282	PCA(n_components=28)
ExtraTreeClassifier	0.933641975	0.927495982	0.933641975	0.930359532	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.932461874	0.928230607	0.932461874	0.930251347	SparsePCA(n_components=28)

KNeighborsClassifier	0.9458061	0.92647585	0.9458061	0.9301090 31	SparsePCA(n_components=28)
SelfTrainingClassifier	0.9458061	0.92647585	0.9458061	0.9301090 31	SparsePCA(n_components=28)
ExtraTreeClassifier	0.9324618 74	0.9278850 72	0.9324618 74	0.9300629 88	SelectKBest (f_classif, k=28)
ExtraTreeClassifier	0.9324618 74	0.9278850 72	0.9324618 74	0.9300629 88	SelectKBest (f_regression, k=28)
ExtraTreeClassifier	0.9317356 57	0.9279169 55	0.9317356 57	0.9297518 28	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.9478031 95	0.9306798 96	0.9478031 95	0.9296964 52	SparsePCA(n_components=28)
ExtraTreeClassifier	0.9303740 02	0.9289644 93	0.9303740 02	0.9296595 35	PCA(n_components=28)
ExtraTreeClassifier	0.9326434 28	0.9269178 7	0.9326434 28	0.9296088 39	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
HistGradientBoostingClassifier	0.9472585 33	0.9289802 57	0.9472585 33	0.9296021 84	MiniBatchSparsePCA(n_components=28)
LGBMClassifier	0.9476216 41	0.9300429 23	0.9476216 41	0.9294519 18	SparsePCA(n_components=28)
ExtraTreeClassifier	0.9318264 34	0.9272590 49	0.9318264 34	0.9294368 19	RFECV (estimator=SVC(), min_features_to_select=28)
KNeighborsClassifier	0.9449891 07	0.9242638 34	0.9449891 07	0.9288788 28	SelectKBest (mutual_info_classif, k=28)
SelfTrainingClassifier	0.9449891 07	0.9242638 34	0.9449891 07	0.9288788 28	SelectKBest (mutual_info_classif, k=28)
KNeighborsClassifier	0.9449891 07	0.9242638 34	0.9449891 07	0.9288788 28	SelectKBest (f_regression, k=28)
SelfTrainingClassifier	0.9449891 07	0.9242638 34	0.9449891 07	0.9288788 28	SelectKBest (f_regression, k=28)
KNeighborsClassifier	0.9449891 07	0.9242638 34	0.9449891 07	0.9288788 28	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.9449891 07	0.9242638 34	0.9449891 07	0.9288788 28	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.9449891 07	0.9242638 34	0.9449891 07	0.9288788 28	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)

SelfTrainingClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFE (estimator=LogisticRegression(), n_features_to_select=28)
SelfTrainingClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFE (estimator=LogisticRegression(), n_features_to_select=28)
KNeighborsClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFE (estimator=SVC(), n_features_to_select=28)
SelfTrainingClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFE (estimator=SVC(), n_features_to_select=28)
KNeighborsClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFE (estimator=MLPClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFE (estimator=MLPClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
SelfTrainingClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
KNeighborsClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFECV (estimator=SVC(), min_features_to_select=28)
SelfTrainingClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFECV (estimator=SVC(), min_features_to_select=28)
KNeighborsClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.944989107	0.924263834	0.944989107	0.928878828	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.944989107	0.924263834	0.944989107	0.928878828	PCA(n_components=28)

SelfTrainingClassifier	0.944989107	0.924263834	0.944989107	0.928878828	PCA(n_components=28)
DecisionTreeClassifier	0.930464779	0.92703786	0.930464779	0.928695287	SelectKBest (f_regression, k=28)
DecisionTreeClassifier	0.930464779	0.92703786	0.930464779	0.928695287	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.94426289	0.923275637	0.94426289	0.928646689	MiniBatchSparsePCA(n_components=28)
SelfTrainingClassifier	0.94426289	0.923275637	0.94426289	0.928646689	MiniBatchSparsePCA(n_components=28)
DecisionTreeClassifier	0.930283224	0.927081246	0.930283224	0.928633652	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
ExtraTreeClassifier	0.930464779	0.926920516	0.930464779	0.928632826	RFE (estimator=SVC(), n_features_to_select=28)
HistGradientBoostingClassifier	0.947530864	0.929646651	0.947530864	0.92860779	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.94489833	0.923842093	0.94489833	0.928583117	SelectKBest (f_classif, k=28)
SelfTrainingClassifier	0.94489833	0.923842093	0.94489833	0.928583117	SelectKBest (f_classif, k=28)
ExtraTreeClassifier	0.930827887	0.926481259	0.930827887	0.92856388	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.92854199	SelectKBest (f_classif, k=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.92854199	SelectKBest (mutual_info_classif, k=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.92854199	SelectKBest (f_regression, k=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.92854199	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.92854199	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.92854199	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.92854199	RFE (estimator=SVC(), n_features_to_select=28)

LGBMClassifier	0.948075527	0.932105571	0.948075527	0.928541999	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.928541999	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.928541999	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.928541999	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.928541999	RFECV (estimator=SVC(), min_features_to_select=28)
LGBMClassifier	0.948075527	0.932105571	0.948075527	0.928541999	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.929920116	0.927169644	0.929920116	0.928509514	RFE (estimator=SVC(), n_features_to_select=28)
DecisionTreeClassifier	0.930192447	0.926809695	0.930192447	0.928447297	RFE (estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.930646333	0.926405513	0.930646333	0.928440364	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.93010167	0.926773005	0.93010167	0.928385488	SelectKBest (mutual_info_classif, k=28)
ExtraTreeClassifier	0.931009441	0.925963337	0.931009441	0.92836432	SelectKBest (mutual_info_classif, k=28)
DecisionTreeClassifier	0.928467683	0.928007061	0.928467683	0.928236424	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.947530864	0.929652968	0.947530864	0.928067542	SelectKBest (f_classif, k=28)
DecisionTreeClassifier	0.929375454	0.926719926	0.929375454	0.928015677	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.929647785	0.926473231	0.929647785	0.928014448	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
MLPClassifier	0.945896877	0.92475984	0.945896877	0.927977701	SparsePCA(n_components=28)
ExtraTreeClassifier	0.930374002	0.925576081	0.930374002	0.927868374	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.92919939	0.926531066	0.92919939	0.927830627	SelectKBest (f_classif, k=28)

DecisionTreeClassifier	0.92901234 6	0.92669664 2	0.92901234 6	0.92783050 7	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.94753086 4	0.92968210 4	0.94753086 4	0.92779243 1	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	SelectKBest (mutual_info_classif, k=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	SelectKBest (f_regression, k=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	RFE (estimator=SVC(), n_features_to_select=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	RFECV (estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.94571532 3	0.92418779 7	0.94571532 3	0.92773601 6	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
MLPClassifier	0.94689542 5	0.92714316 7	0.94689542 5	0.92767555 4	SelectKBest (mutual_info_classif, k=28)
MLPClassifier	0.94725853 3	0.92847564 1	0.94725853 3	0.92748710 4	SelectKBest (f_regression, k=28)
LabelSpreading	0.94589687 7	0.92429483 4	0.94589687 7	0.92745918 7	SelectKBest (mutual_info_classif, k=28)
LabelSpreading	0.94589687 7	0.92429483 4	0.94589687 7	0.92745918 7	SelectKBest (f_regression, k=28)

LabelSpreading	0.945896877	0.924294834	0.945896877	0.927459187	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelSpreading	0.945896877	0.924294834	0.945896877	0.927459187	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.945896877	0.924294834	0.945896877	0.927459187	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.945896877	0.924294834	0.945896877	0.927459187	RFE (estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.945896877	0.924294834	0.945896877	0.927459187	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LabelSpreading	0.945896877	0.924294834	0.945896877	0.927459187	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelSpreading	0.945896877	0.924294834	0.945896877	0.927459187	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
LabelSpreading	0.945896877	0.924294834	0.945896877	0.927459187	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.945896877	0.924294834	0.945896877	0.927459187	RFECV (estimator=SVC(), min_features_to_select=28)
LabelSpreading	0.945896877	0.924294834	0.945896877	0.927459187	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.9291939	0.925698407	0.9291939	0.927392068	SparsePCA(n_components=28)
MLPClassifier	0.947076979	0.927709082	0.947076979	0.927376234	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
MLPClassifier	0.947076979	0.927709082	0.947076979	0.927376234	PCA(n_components=28)
LabelPropagation	0.946259986	0.925010666	0.946259986	0.927284868	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.947803195	0.931387262	0.947803195	0.927254025	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.945533769	0.923349507	0.945533769	0.927234867	SelectKBest (f_classif, k=28)
LabelPropagation	0.945533769	0.923349507	0.945533769	0.927234867	PCA(n_components=28)
HistGradientBoostingClassifier	0.947530864	0.929807973	0.947530864	0.927231933	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)

LabelSpreading	0.946532317	0.92571404	0.946532317	0.9271810218	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.94734931	0.928954	0.94734931	0.9271220558	SelectKBest (mutual_info_classif, k=28)
MLPClassifier	0.946169208	0.924622303	0.946169208	0.927094716	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947167756	0.928034475	0.947167756	0.927012282	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
LabelSpreading	0.945715323	0.923421816	0.945715323	0.926950994	SelectKBest (f_classif, k=28)
LabelSpreading	0.945715323	0.923421816	0.945715323	0.926950994	PCA(n_components=28)
DecisionTreeClassifier	0.928104575	0.925637856	0.928104575	0.926845425	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.927832244	0.925775018	0.927832244	0.92678583	RFECV (estimator=SVC(), min_features_to_select=28)
LabelSpreading	0.945442992	0.922697017	0.945442992	0.926784395	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.94798475	0.933247201	0.94798475	0.926781411	SelectKBest (f_classif, k=28)
MLPClassifier	0.946259986	0.924479391	0.946259986	0.926603582	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.947167756	0.928015692	0.947167756	0.926585051	RFE (estimator=MLPClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.946623094	0.925642992	0.946623094	0.926543165	RFECV (estimator=SVC(), min_features_to_select=28)
DecisionTreeClassifier	0.92765069	0.925468147	0.92765069	0.926539632	RFE (estimator=MLPClassifier(), n_features_to_select=28)
MLPClassifier	0.945896877	0.923415571	0.945896877	0.926521674	RFECV (estimator=SVC(), min_features_to_select=28)
VotingClassifier	0.947167756	0.928018712	0.947167756	0.926440857	SparsePCA(n_components=28)
MLPClassifier	0.945896877	0.923281226	0.945896877	0.926384504	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.944353667	0.920356573	0.944353667	0.926250403	MiniBatchSparsePCA(n_components=28)

HistGradientBoostingClassifier	0.946986202	0.927071376	0.946986202	0.926188077	SelectKBest (f_regression, k=28)
MLPClassifier	0.947440087	0.929848389	0.947440087	0.926014724	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947349314	0.929240184	0.947349314	0.925961444	RFE (estimator=SVC(), n_features_to_select=28)
MLPClassifier	0.946895425	0.926546924	0.946895425	0.925842572	RFE (estimator=MLPClassifier(), n_features_to_select=28)
MLPClassifier	0.947076979	0.927547305	0.947076979	0.925801725	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947440087	0.930094184	0.947440087	0.925714852	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.927469136	0.923818913	0.927469136	0.925590868	PCA(n_components=28)
VotingClassifier	0.946895425	0.926490453	0.946895425	0.925547185	PCA(n_components=28)
LinearDiscriminantAnalysis	0.940813362	0.917336582	0.940813362	0.925544144	SelectKBest (f_classif, k=28)
LinearDiscriminantAnalysis	0.940813362	0.917336582	0.940813362	0.925544144	SelectKBest (mutual_info_classif, k=28)
LinearDiscriminantAnalysis	0.940813362	0.917336582	0.940813362	0.925544144	SelectKBest (f_regression, k=28)
LinearDiscriminantAnalysis	0.940813362	0.917336582	0.940813362	0.925544144	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.940813362	0.917336582	0.940813362	0.925544144	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.940813362	0.917336582	0.940813362	0.925544144	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.940813362	0.917336582	0.940813362	0.925544144	RFE (estimator=SVC(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.940813362	0.917336582	0.940813362	0.925544144	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.940813362	0.917336582	0.940813362	0.925544144	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.940813362	0.917336582	0.940813362	0.925544144	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)

LinearDiscriminantAnalysis	0.940813362	0.91733658 2	0.940813362	0.925544144	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.940813362	0.91733658 2	0.940813362	0.925544144	RFECV (estimator=SVC(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.940813362	0.91733658 2	0.940813362	0.925544144	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.940813362	0.91733658 2	0.940813362	0.925544144	PCA(n_components=28)
LinearDiscriminantAnalysis	0.940813362	0.91733658 2	0.940813362	0.925544144	SparsePCA(n_components=28)
LinearDiscriminantAnalysis	0.940813362	0.91733658 2	0.940813362	0.925544144	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.947621641	0.93199786 1	0.947621641	0.925515953	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
VotingClassifier	0.946986202	0.92701454 7	0.946986202	0.925450833	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.947440087	0.93041466 4	0.947440087	0.925411116	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.947621641	0.93228765 2	0.947621641	0.925362152	RFE (estimator=LogisticRegression(), n_features_to_select=28)
VotingClassifier	0.947621641	0.93228765 2	0.947621641	0.925362152	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.947621641	0.93228765 2	0.947621641	0.925362152	SparsePCA(n_components=28)
GradientBoostingClassifier	0.947803195	0.93460447 3	0.947803195	0.925311326	SparsePCA(n_components=28)
VotingClassifier	0.947530864	0.93142782 9	0.947530864	0.925309955	SelectKBest (f_regression, k=28)
VotingClassifier	0.947440087	0.93060973 3	0.947440087	0.925257776	SelectKBest (mutual_info_classif, k=28)
VotingClassifier	0.947440087	0.93060973 3	0.947440087	0.925257776	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
VotingClassifier	0.947349311	0.92983038 1	0.947349311	0.925205614	SelectKBest (f_classif, k=28)
MLPClassifier	0.947349311	0.92983038 1	0.947349311	0.925205614	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)

VotingClassifier	0.94734931	0.929830381	0.94734931	0.925205614	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
VotingClassifier	0.947530864	0.93170161	0.947530864	0.925155388	RFECV (estimator=SVC(), min_features_to_select=28)
HistGradientBoostingClassifier	0.947258533	0.929087073	0.947258533	0.925153469	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.947440087	0.930832776	0.947440087	0.925103442	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.947621641	0.932993213	0.947621641	0.92505154	PCA(n_components=28)
VotingClassifier	0.94734931	0.930007162	0.94734931	0.925051512	RFE (estimator=SVC(), n_features_to_select=28)
VotingClassifier	0.94734931	0.930007162	0.94734931	0.925051512	RFE (estimator=MLPClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.947258533	0.929377901	0.947258533	0.924844732	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.946713871	0.925154095	0.946713871	0.924534988	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
MLPClassifier	0.946986202	0.927228956	0.946986202	0.924378454	RFE (estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.94734931	0.931830962	0.94734931	0.924105551	RFE (estimator=MLPClassifier(), n_features_to_select=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	SelectKBest (f_classif, k=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	SelectKBest (mutual_info_classif, k=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	SelectKBest (f_regression, k=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	RFE (estimator=LogisticRegression(), n_features_to_select=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	RFE (estimator=SVC(), n_features_to_select=28)

SVC	0.947712418	0.940082312	0.947712418	0.923981746	RFE (estimator=MLPClassifier(), n_features_to_select=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	RFECV (estimator=SVC(), min_features_to_select=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.947712418	0.940082312	0.947712418	0.923981746	PCA(n_components=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	SelectKBest (f_classif, k=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	SelectKBest (mutual_info_classif, k=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	SelectKBest (f_regression, k=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	RFE (estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	RFE (estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	RFECV (estimator=SVC(), min_features_to_select=28)

GradientBoostingClassifier	0.947258533	0.931043244	0.947258533	0.923894037	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.946986202	0.927669044	0.946986202	0.923743498	PCA(n_components=28)
AdaBoostClassifier	0.946986202	0.927669044	0.946986202	0.923743498	SparsePCA(n_components=28)
SVC	0.947167756	0.930201236	0.947167756	0.923681979	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.945079884	0.916965642	0.945079884	0.923458583	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.937000726	0.914087937	0.937000726	0.923310808	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	SelectKBest (f_classif, k=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	SelectKBest (mutual_info_classif, k=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	SelectKBest (f_regression, k=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	RFE (estimator=LogisticRegression(), n_features_to_select=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	RFE (estimator=SVC(), n_features_to_select=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	RFE (estimator=MLPClassifier(), n_features_to_select=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	RFECV (estimator=SVC(), min_features_to_select=28)

GaussianNB	0.933278867	0.915627676	0.933278867	0.923248277	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.946230944	0.923938251	0.946230944	0.923221193	MiniBatchSparsePCA(n_components=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	SelectKBest (f_classif, k=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	SelectKBest (mutual_info_classif, k=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	SelectKBest (f_regression, k=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	RFE (estimator=SVC(), n_features_to_select=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	RFECV (estimator=SVC(), min_features_to_select=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.945987654	0.919504705	0.945987654	0.923192366	PCA(n_components=28)
PassiveAggressiveClassifier	0.934368192	0.914904856	0.934368192	0.923151656	SelectKBest (f_regression, k=28)
LogisticRegression	0.945896877	0.918973267	0.945896877	0.923142327	SparsePCA(n_components=28)

LogisticRegression	0.945896877	0.918973267	0.945896877	0.923142321	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.947076979	0.930160867	0.947076979	0.923140754	RFE (estimator=LogisticRegression(), n_features_to_select=28)
SGDClassifier	0.946441541	0.922261232	0.946441541	0.923121987	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.944172113	0.913837884	0.944172113	0.922799143	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
SGDClassifier	0.944989107	0.914773548	0.944989107	0.922642474	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	SelectKBest (f_classif, k=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	SelectKBest (mutual_info_classif, k=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	SelectKBest (f_regression, k=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	RFE (estimator=LogisticRegression(), n_features_to_select=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	RFE (estimator=SVC(), n_features_to_select=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	RFE (estimator=MLPClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	RFECV (estimator=SVC(), min_features_to_select=28)
AdaBoostClassifier	0.946623094	0.923629764	0.946623094	0.922564857	RFECV (estimator=MLPClassifier(), min_features_to_select=28)

RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	SelectKBest (f_classif, k=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	SelectKBest (f_classif, k=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	SelectKBest (mutual_info_classif, k=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	SelectKBest (mutual_info_classif, k=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	SelectKBest (f_regression, k=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	SelectKBest (f_regression, k=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	RFE (estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	RFE (estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	RFE (estimator=SVC(), n_features_to_select=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	RFE (estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	RFE (estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	RFE (estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)

RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	RFECV (estimator=SVC(), min_features_to_select=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	RFECV (estimator=SVC(), min_features_to_select=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	PCA(n_components=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	PCA(n_components=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	SparsePCA(n_components=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	SparsePCA(n_components=28)
RidgeClassifier	0.946532317	0.922464877	0.946532317	0.922516302	MiniBatchSparsePCA(n_components=28)
RidgeClassifierCV	0.946532317	0.922464877	0.946532317	0.922516302	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.939996369	0.911808329	0.939996369	0.922488314	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.946986202	0.931119080	0.946986202	0.922422166	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.945987654	0.917648809	0.945987654	0.922388953	SelectKBest (mutual_info_classif, k=28)
SGDClassifier	0.944898330	0.912986482	0.944898330	0.922122479	SparsePCA(n_components=28)

SGDClassifier	0.946350763	0.919649454	0.946350763	0.922086082	PCA(n_components=28)
SGDClassifier	0.946986202	0.934015275	0.946986202	0.92208062	RFECV (estimator=SVC(), min_features_to_select=28)
SGDClassifier	0.946532317	0.921824246	0.946532317	0.922013293	RFE (estimator=SVC(), n_features_to_select=28)
SGDClassifier	0.946804648	0.927778887	0.946804648	0.92198571	SelectKBest (f_classif, k=28)
SGDClassifier	0.946804648	0.927778887	0.946804648	0.92198571	RFE (estimator=MLPClassifier(), n_features_to_select=28)
SGDClassifier	0.94644154	0.920424232	0.94644154	0.921965568	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
SGDClassifier	0.946713871	0.925379226	0.946713871	0.921938259	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
SGDClassifier	0.946895425	0.932154583	0.946895425	0.921860936	SelectKBest (f_regression, k=28)
PassiveAggressiveClassifier	0.934277415	0.912490946	0.934277415	0.92176014	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
SGDClassifier	0.946259986	0.914176584	0.946259986	0.921186929	SelectKBest (mutual_info_classif, k=28)
PassiveAggressiveClassifier	0.943990559	0.908322579	0.943990559	0.921159041	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.941176471	0.907637684	0.941176471	0.920843218	RFE (estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.933460421	0.90916622	0.933460421	0.919744128	SelectKBest (f_classif, k=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	SelectKBest (f_classif, k=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	SelectKBest (mutual_info_classif, k=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	SelectKBest (f_regression, k=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)

QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	RFE (estimator=LogisticRegression(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	RFE (estimator=SVC(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	RFE (estimator=MLPClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	RFECV (estimator=SVC(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	PCA(n_components=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	SparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.924473493	0.915096172	0.924473493	0.919542813	MiniBatchSparsePCA(n_components=28)
GaussianNB	0.923474946	0.915197052	0.923474946	0.919152176	PCA(n_components=28)
GaussianNB	0.923474946	0.915197052	0.923474946	0.919152176	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.928013798	0.909170208	0.928013798	0.917797214	RFE (estimator=MLPClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.930010893	0.906277607	0.930010893	0.917072633	SparsePCA(n_components=28)
GaussianNB	0.917392883	0.916322241	0.917392883	0.916854841	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.89778504	0.917804808	0.89778504	0.907123346	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.869825708	0.919012602	0.869825708	0.891516355	RFE (estimator=LogisticRegression(), n_features_to_select=28)

PassiveAggressiveClassifier	0.874183007	0.901151606	0.874183007	0.887190995	RFECV (estimator=SVC(), min_features_to_select=28)
PassiveAggressiveClassifier	0.798202614	0.918258115	0.798202614	0.847686798	PCA(n_components=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	SelectKBest (f_classif, k=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	SelectKBest (mutual_info_classif, k=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	SelectKBest (f_regression, k=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	RFE (estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	RFE (estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	RFE (estimator=MLPClassifier(), n_features_to_select=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	RFECV (estimator=SVC(), min_features_to_select=28)
BernoulliNB	0.770243283	0.925449534	0.770243283	0.830614678	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
BernoulliNB	0.738108206	0.915997264	0.738108206	0.808358528	MiniBatchSparsePCA(n_components=28)
BernoulliNB	0.735929557	0.922012192	0.735929557	0.807337237	PCA(n_components=28)
BernoulliNB	0.735929557	0.922012192	0.735929557	0.807337237	SparsePCA(n_components=28)

PassiveAggressiveClassifier	0.5515613655	0.8812540733	0.5515613655	0.6713873999	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.4470769799	0.8699095008	0.4470769799	0.5802078933	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)

APPENDIX 4: VULNERABILITY FUNCTION LEVEL DATA RESULTS IN 2002 TO 2016

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
RandomForestClassifier	0.957143	0.94753	0.957143	0.947079	PCA(n_components=28)
RandomForestClassifier	0.957086	0.94745	0.957086	0.947523	RFE (estimator=MLPClassifier(), n_features_to_select=28)
RandomForestClassifier	0.957029	0.94733	0.957029	0.94732	SelectKBest (mutual_info_classif, k=28)
RandomForestClassifier	0.957029	0.94733	0.957029	0.94732	RFE (estimator=SVC(), n_features_to_select=28)
StackingClassifier	0.956914	0.947139	0.956914	0.946003	RFE (estimator=SVC(), n_features_to_select=28)
StackingClassifier	0.9568	0.946901	0.9568	0.94586	RFECV (estimator=SVC(), min_features_to_select=28)
StackingClassifier	0.9568	0.946881	0.9568	0.946034	MiniBatchSparsePCA(n_components=28)
RandomForestClassifier	0.9568	0.94688	0.9568	0.946822	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.956743	0.946814	0.956743	0.945582	SelectKBest (f_regression, k=28)
RandomForestClassifier	0.956743	0.946802	0.956743	0.946996	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
RandomForestClassifier	0.956629	0.946666	0.956629	0.947231	SelectKBest (f_regression, k=28)
RandomForestClassifier	0.956629	0.946619	0.956629	0.947018	RFE (estimator=LogisticRegression(), n_features_to_select=28)
StackingClassifier	0.956629	0.946554	0.956629	0.945557	RFE (estimator=MLPClassifier(), n_features_to_select=28)
StackingClassifier	0.956629	0.946554	0.956629	0.945557	SparsePCA(n_components=28)
RandomForestClassifier	0.956571	0.946488	0.956571	0.946813	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
StackingClassifier	0.956571	0.946454	0.956571	0.945337	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
StackingClassifier	0.956571	0.946438	0.956571	0.945455	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.956514	0.946304	0.956514	0.945531	RFE (estimator=LogisticRegression(), n_features_to_select=28)

StackingClassifier	0.956514	0.946297	0.956514	0.945648	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.956514	0.946294	0.956514	0.945764	PCA(n_components=28)
RandomForestClassifier	0.956457	0.946255	0.956457	0.946564	SelectKBest (f_classif, k=28)
RandomForestClassifier	0.956343	0.946064	0.956343	0.946533	RFECV (estimator=SVC(), min_features_to_select=28)
RandomForestClassifier	0.956343	0.946041	0.956343	0.946424	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
StackingClassifier	0.956343	0.945952	0.956343	0.945228	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
RandomForestClassifier	0.956286	0.945872	0.956286	0.945993	MiniBatchSparsePCA(n_components=28)
StackingClassifier	0.956286	0.945829	0.956286	0.945245	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
RandomForestClassifier	0.956114	0.945733	0.956114	0.946578	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
StackingClassifier	0.956229	0.945711	0.956229	0.945203	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
StackingClassifier	0.956171	0.945596	0.956171	0.944923	SelectKBest (f_classif, k=28)
RandomForestClassifier	0.956	0.945509	0.956	0.946386	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.956114	0.945479	0.956114	0.944761	SelectKBest (mutual_info_classif, k=28)
RandomForestClassifier	0.955943	0.945209	0.955943	0.945569	SparsePCA(n_components=28)
BaggingClassifier	0.955086	0.944321	0.955086	0.94618	RFECV (estimator=SVC(), min_features_to_select=28)
BaggingClassifier	0.955143	0.944287	0.955143	0.946013	RFE (estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.954857	0.943874	0.954857	0.945801	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
BaggingClassifier	0.954857	0.943843	0.954857	0.945749	RFE (estimator=LogisticRegression(), n_features_to_select=28)
BaggingClassifier	0.954743	0.943524	0.954743	0.945399	RFE (estimator=MLPClassifier(), n_features_to_select=28)

BaggingClassifier	0.954743	0.943434	0.954743	0.945237	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.954914	0.943403	0.954914	0.944756	SelectKBest (mutual_info_classif, k=28)
XGBClassifier	0.955086	0.943326	0.955086	0.941647	PCA(n_components=28)
BaggingClassifier	0.954629	0.943054	0.954629	0.94477	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
BaggingClassifier	0.954514	0.942939	0.954514	0.944797	SelectKBest (f_classif, k=28)
BaggingClassifier	0.954343	0.942715	0.954343	0.944726	SelectKBest (f_regression, k=28)
BaggingClassifier	0.954343	0.94265	0.954343	0.944616	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
BaggingClassifier	0.954229	0.942581	0.954229	0.944697	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
BaggingClassifier	0.954514	0.942564	0.954514	0.944066	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.954171	0.942465	0.954171	0.9446	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
BaggingClassifier	0.954286	0.942404	0.954286	0.944297	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.954114	0.94239	0.954114	0.936887	PCA(n_components=28)
XGBClassifier	0.954571	0.941992	0.954571	0.941692	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.9536	0.941967	0.9536	0.934812	SelectKBest (f_regression, k=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	SelectKBest(f_classif, k=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	SelectKBest(mutual_info_classif, k=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	SelectKBest(f_regression, k=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	RFE(estimator=LogisticRegression(), n_features_to_select=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	RFE(estimator=SVC(), n_features_to_select=28)

XGBClassifier	0.954514	0.941931	0.954514	0.940269	RFE(estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	RFECV (estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.954514	0.941931	0.954514	0.940269	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
XGBClassifier	0.954457	0.941702	0.954457	0.941344	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.9532	0.94054	0.9532	0.943116	PCA(n_components=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	SelectKBest (f_classif, k=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	SelectKBest (mutual_info_classif, k=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	SelectKBest (f_regression, k=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	RFE (estimator=DecisionTreeClassifier(), n_features_to_select=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	RFE (estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	RFE (estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	RFE (estimator=SVC(), n_features_to_select=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	RFE (estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	RFECV (estimator=DecisionTreeClassifier(), min_features_to_select=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	RFECV (estimator=RandomForestClassifier(), min_features_to_select=28)

LGBMClassifier	0.953543	0.940371	0.953543	0.935582	RFECV (estimator=LogisticRegression(), min_features_to_select=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	RFECV (estimator=SVC(), min_features_to_select=28)
LGBMClassifier	0.953543	0.940371	0.953543	0.935582	RFECV (estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.952686	0.939932	0.952686	0.93158	MiniBatchSparsePCA(n_components=28)
SVC	0.952571	0.939813	0.952571	0.931106	SelectKBest (f_classif, k=28)
SVC	0.952571	0.939813	0.952571	0.931106	SelectKBest (mutual_info_classif, k=28)
SVC	0.952571	0.939813	0.952571	0.931106	SelectKBest(f_regression, k=28)
SVC	0.952571	0.939813	0.952571	0.931106	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SVC	0.952571	0.939813	0.952571	0.931106	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SVC	0.952571	0.939813	0.952571	0.931106	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SVC	0.952571	0.939813	0.952571	0.931106	RFE(estimator=SVC(), n_features_to_select=28)
SVC	0.952571	0.939813	0.952571	0.931106	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SVC	0.952571	0.939813	0.952571	0.931106	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.952571	0.939813	0.952571	0.931106	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SVC	0.952571	0.939813	0.952571	0.931106	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SVC	0.952571	0.939813	0.952571	0.931106	RFECV(estimator=SVC(), min_features_to_select=28)
SVC	0.952571	0.939813	0.952571	0.931106	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.952571	0.939813	0.952571	0.931106	PCA(n_components=28)
HistGradientBoostingClassifier	0.9532	0.939795	0.9532	0.934111	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.953314	0.93971	0.953314	0.93482	MiniBatchSparsePCA(n_components=28)
SVC	0.952686	0.939549	0.952686	0.931682	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.953257	0.939094	0.953257	0.935053	SelectKBest(mutual_info_classif, k=28)

HistGradientBoostingClassifier	0.953086	0.93897	0.953086	0.934043	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.953143	0.938834	0.953143	0.934535	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.952457	0.93864	0.952457	0.930836	MiniBatchSparsePCA(n_components=28)
RidgeClassifierCV	0.952457	0.93864	0.952457	0.930836	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.952914	0.938194	0.952914	0.933569	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.953086	0.937964	0.953086	0.935299	SparsePCA(n_components=28)
LGBMClassifier	0.952971	0.937805	0.952971	0.934432	PCA(n_components=28)
HistGradientBoostingClassifier	0.952857	0.937778	0.952857	0.933535	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.952571	0.93773	0.952571	0.931719	SelectKBest(mutual_info_classif, k=28)
GradientBoostingClassifier	0.952571	0.93773	0.952571	0.931719	SelectKBest(f_regression, k=28)
GradientBoostingClassifier	0.952571	0.93773	0.952571	0.931719	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.952571	0.93773	0.952571	0.931719	RFE(estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.952571	0.93773	0.952571	0.931719	RFECV(estimator=SVC(), min_features_to_select=28)
LGBMClassifier	0.952971	0.937595	0.952971	0.93479	SparsePCA(n_components=28)
VotingClassifier	0.952743	0.937441	0.952743	0.932995	RFE(estimator=SVC(), n_features_to_select=28)
VotingClassifier	0.952743	0.937441	0.952743	0.932995	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.9528	0.937373	0.9528	0.933502	RFE(estimator=LogisticRegression(), n_features_to_select=28)
VotingClassifier	0.952743	0.937332	0.952743	0.93309	SelectKBest(mutual_info_classif, k=28)
RidgeClassifier	0.95224	0.937237	0.95224	0.930909	SelectKBest(f_classif, k=28)
RidgeClassifierCV	0.95224	0.937237	0.95224	0.930909	SelectKBest(f_classif, k=28)
RidgeClassifier	0.95224	0.937237	0.95224	0.930909	SelectKBest(mutual_info_classif, k=28)
RidgeClassifierCV	0.95224	0.937237	0.95224	0.930909	SelectKBest(mutual_info_classif, k=28)
RidgeClassifier	0.95224	0.937237	0.95224	0.930909	SelectKBest(f_regression, k=28)
RidgeClassifierCV	0.95224	0.937237	0.95224	0.930909	SelectKBest(f_regression, k=28)

RidgeClassifier	0.9524	0.937237	0.9524	0.930909	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	RFECV(estimator=MLPClassifier(), min_features_to_select=28)

RidgeClassifier	0.9524	0.937237	0.9524	0.930909	PCA(n_components=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	PCA(n_components=28)
RidgeClassifier	0.9524	0.937237	0.9524	0.930909	SparsePCA(n_components=28)
RidgeClassifierCV	0.9524	0.937237	0.9524	0.930909	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.952743	0.93714	0.952743	0.93328	SelectKBest(f_classif, k=28)
GradientBoostingClassifier	0.952514	0.937042	0.952514	0.931687	SelectKBest(f_classif, k=28)
GradientBoostingClassifier	0.952514	0.937042	0.952514	0.931687	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.952514	0.937042	0.952514	0.931687	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.952514	0.937042	0.952514	0.931687	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.952514	0.937042	0.952514	0.931687	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.952514	0.937042	0.952514	0.931687	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.952514	0.937042	0.952514	0.931687	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.952514	0.937042	0.952514	0.931687	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.952686	0.936988	0.952686	0.932962	SelectKBest(f_regression, k=28)
VotingClassifier	0.952686	0.936988	0.952686	0.932962	RFE(estimator=LogisticRegression(), n_features_to_select=28)
VotingClassifier	0.952686	0.936988	0.952686	0.932962	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.9528	0.936843	0.9528	0.934328	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.952686	0.936809	0.952686	0.933152	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.940743	0.936796	0.940743	0.93867	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.9528	0.936776	0.9528	0.934507	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.952686	0.936731	0.952686	0.933246	RFE(estimator=MLPClassifier(), n_features_to_select=28)

VotingClassifier	0.952629	0.936636	0.952629	0.932833	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
VotingClassifier	0.952629	0.936549	0.952629	0.932928	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.952629	0.936549	0.952629	0.932928	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
ExtraTreeClassifier	0.941543	0.936463	0.941543	0.938832	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.952457	0.936384	0.952457	0.931655	RFE(estimator=MLPClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.952571	0.936362	0.952571	0.932607	RFECV(estimator=SVC(), min_features_to_select=28)
DecisionTreeClassifier	0.940057	0.936344	0.940057	0.938117	MiniBatchSparsePCA(n_components=28)
ExtraTreeClassifier	0.941714	0.936333	0.941714	0.93883	RFECV(estimator=SVC(), min_features_to_select=28)
ExtraTreeClassifier	0.941886	0.936204	0.941886	0.938828	SelectKBest(f_classif, k=28)
VotingClassifier	0.952571	0.936194	0.952571	0.9328	RFECV(estimator=SVC(), min_features_to_select=28)
VotingClassifier	0.952571	0.936123	0.952571	0.932895	SelectKBest(f_classif, k=28)
VotingClassifier	0.952571	0.936123	0.952571	0.932895	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
VotingClassifier	0.952571	0.936123	0.952571	0.932895	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.940286	0.936089	0.940286	0.938079	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.939371	0.935972	0.939371	0.937604	SparsePCA(n_components=28)
ExtraTreeClassifier	0.941543	0.935964	0.941543	0.938549	PCA(n_components=28)
ExtraTreeClassifier	0.941429	0.935836	0.941429	0.938429	SelectKBest(f_regression, k=28)
ExtraTreeClassifier	0.9412	0.935798	0.9412	0.938311	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
VotingClassifier	0.952514	0.935766	0.952514	0.932767	MiniBatchSparsePCA(n_components=28)
ExtraTreeClassifier	0.941257	0.935753	0.941257	0.93831	RFE(estimator=SVC(), n_features_to_select=28)
DecisionTreeClassifier	0.938514	0.935753	0.938514	0.937091	RFECV(estimator=SVC(), min_features_to_select=28)
GradientBoostingClassifier	0.9524	0.93563	0.9524	0.931723	PCA(n_components=28)
DecisionTreeClassifier	0.938343	0.935609	0.938343	0.936935	SelectKBest(f_regression, k=28)

VotingClassifier	0.952514	0.935504	0.952514	0.933333	SparsePCA(n_components=28)
DecisionTreeClassifier	0.938229	0.935417	0.938229	0.93678	SelectKBest(f_classif, k=28)
DecisionTreeClassifier	0.939029	0.93539	0.939029	0.937134	PCA(n_components=28)
DecisionTreeClassifier	0.937486	0.935334	0.937486	0.936386	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.9392	0.935246	0.9392	0.937133	SelectKBest(mutual_info_classif, k=28)
ExtraTreeClassifier	0.938514	0.935244	0.938514	0.93682	RFE(estimator=MLPClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.9404	0.935202	0.9404	0.937636	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.938286	0.935149	0.938286	0.936663	RFE(estimator=LogisticRegression(), n_features_to_select=28)
DecisionTreeClassifier	0.938229	0.935125	0.938229	0.936624	RFE(estimator=MLPClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.937429	0.935092	0.937429	0.936232	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.937714	0.935061	0.937714	0.93635	SparsePCA(n_components=28)
DecisionTreeClassifier	0.937371	0.934996	0.937371	0.936154	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.937314	0.934974	0.937314	0.936115	RFE(estimator=SVC(), n_features_to_select=28)
MLPClassifier	0.9524	0.934971	0.9524	0.934877	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
VotingClassifier	0.9524	0.934947	0.9524	0.9327	PCA(n_components=28)
ExtraTreeClassifier	0.94	0.934944	0.94	0.937 32	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.937657	0.934891	0.937657	0.936234	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.937143	0.934758	0.937143	0.935921	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.952286	0.93456	0.952286	0.934892	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.951714	0.934552	0.951714	0.937 23	SparsePCA(n_components=28)
MLPClassifier	0.952343	0.934521	0.952343	0.932857	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.9376	0.934425	0.9376	0.93596	SelectKBest(mutual_info_classif, k=28)

LabelPropagation	0.952171	0.93441	0.952171	0.93541	MiniBatchSparsePCA(n_components=28)
DecisionTreeClassifier	0.937086	0.934366	0.937086	0.935688	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
MLPClassifier	0.952229	0.93422	0.952229	0.934512	RFECV(estimator=SVC(), min_features_to_select=28)
ExtraTreeClassifier	0.938114	0.934192	0.938114	0.936071	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.952229	0.93413	0.952229	0.93425	SelectKBest(f_classif, k=28)
ExtraTreeClassifier	0.938286	0.933966	0.938286	0.936026	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
MLPClassifier	0.952057	0.933868	0.952057	0.935005	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
MLPClassifier	0.952114	0.93374	0.952114	0.934355	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
MLPClassifier	0.952057	0.933649	0.952057	0.93458	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.952171	0.933463	0.952171	0.931595	SparsePCA(n_components=28)
KNeighborsClassifier	0.950629	0.933442	0.950629	0.937515	SparsePCA(n_components=28)
SelfTrainingClassifier	0.950629	0.933442	0.950629	0.937515	SparsePCA(n_components=28)
LabelSpreading	0.951829	0.933405	0.951829	0.935279	SparsePCA(n_components=28)
MLPClassifier	0.952	0.933352	0.952	0.934373	SelectKBest(f_regression, k=28)
MLPClassifier	0.952	0.93326	0.952	0.934199	SelectKBest(mutual_info_classif, k=28)
KNeighborsClassifier	0.951086	0.933152	0.951086	0.936673	MiniBatchSparsePCA(n_components=28)
SelfTrainingClassifier	0.951086	0.933152	0.951086	0.936673	MiniBatchSparsePCA(n_components=28)
KNeighborsClassifier	0.950743	0.933146	0.950743	0.9371	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.950743	0.933146	0.950743	0.9371	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.950743	0.933146	0.950743	0.9371	RFE(estimator=SVC(), n_features_to_select=28)
SelfTrainingClassifier	0.950743	0.933146	0.950743	0.9371	RFE(estimator=SVC(), n_features_to_select=28)
KNeighborsClassifier	0.950743	0.933146	0.950743	0.9371	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)

SelfTrainingClassifier	0.950743	0.933146	0.950743	0.9371	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.950743	0.933146	0.950743	0.9371	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SelfTrainingClassifier	0.950743	0.933146	0.950743	0.9371	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
KNeighborsClassifier	0.950743	0.933146	0.950743	0.9371	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.950743	0.933146	0.950743	0.9371	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.950743	0.933146	0.950743	0.9371	PCA(n_components=28)
SelfTrainingClassifier	0.950743	0.933146	0.950743	0.9371	PCA(n_components=28)
MLPClassifier	0.951829	0.93313	0.951829	0.934864	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.9516	0.933066	0.9516	0.935541	SelectKBest(f_classif, k=28)
KNeighborsClassifier	0.950629	0.932952	0.950629	0.937025	SelectKBest(mutual_info_classif, k=28)
SelfTrainingClassifier	0.950629	0.932952	0.950629	0.937025	SelectKBest(mutual_info_classif, k=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	SelectKBest(f_classif, k=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	SelectKBest(mutual_info_classif, k=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	SelectKBest(f_regression, k=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

BernoulliNB	0.771714	0.932946	0.771714	0.834776	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	RFECV(estimator=SVC(), min_features_to_select=28)
BernoulliNB	0.771714	0.932946	0.771714	0.834776	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelPropagation	0.951543	0.932919	0.951543	0.935505	SparsePCA(n_components=28)
KNeighborsClassifier	0.950571	0.932857	0.950571	0.936988	SelectKBest(f_regression, k=28)
SelfTrainingClassifier	0.950571	0.932857	0.950571	0.936988	SelectKBest(f_regression, k=28)
LabelSpreading	0.951486	0.932773	0.951486	0.935469	SelectKBest(f_regression, k=28)
LabelSpreading	0.951486	0.932773	0.951486	0.935469	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.951486	0.932773	0.951486	0.935469	RFE(estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.951486	0.932773	0.951486	0.935469	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelSpreading	0.951486	0.932773	0.951486	0.935469	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.951486	0.932773	0.951486	0.935469	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelSpreading	0.951486	0.932773	0.951486	0.935469	PCA(n_components=28)
LabelPropagation	0.951429	0.932695	0.951429	0.935513	SelectKBest(f_classif, k=28)
LabelSpreading	0.951257	0.932488	0.951257	0.935642	SelectKBest(mutual_info_classif, k=28)
LabelSpreading	0.951257	0.932488	0.951257	0.935642	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	SelectKBest(f_classif, k=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	SelectKBest(mutual_info_classif, k=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	SelectKBest(f_regression, k=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	RFE(estimator=LogisticRegression(), n_features_to_select=28)

AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	RFE(estimator=SVC(), n_features_to_select=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	RFE(estimator=MLPClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	RFECV(estimator=SVC(), min_features_to_select=28)
AdaBoostClassifier	0.952057	0.932485	0.952057	0.931432	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.950343	0.932482	0.950343	0.936838	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.950343	0.932482	0.950343	0.936838	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.950343	0.932482	0.950343	0.936838	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SelfTrainingClassifier	0.950343	0.932482	0.950343	0.936838	RFE(estimator=LogisticRegression(), n_features_to_select=28)
KNeighborsClassifier	0.950343	0.932482	0.950343	0.936838	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.950343	0.932482	0.950343	0.936838	RFE(estimator=MLPClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.950343	0.932482	0.950343	0.936838	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.950343	0.932482	0.950343	0.936838	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.950343	0.932482	0.950343	0.936838	RFECV(estimator=SVC(), min_features_to_select=28)
SelfTrainingClassifier	0.950343	0.932482	0.950343	0.936838	RFECV(estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.951314	0.932415	0.951314	0.935441	SelectKBest(f_regression, k=28)
LabelPropagation	0.951314	0.932415	0.951314	0.935441	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelPropagation	0.951314	0.932415	0.951314	0.935441	RFE(estimator=SVC(), n_features_to_select=28)
LabelPropagation	0.951314	0.932415	0.951314	0.935441	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)

LabelPropagation	0.951314	0.932415	0.951314	0.935441	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.951314	0.932415	0.951314	0.935441	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelPropagation	0.951314	0.932415	0.951314	0.935441	PCA(n_components=28)
LabelPropagation	0.951086	0.932171	0.951086	0.935611	SelectKBest(mutual_info_classif, k=28)
LabelPropagation	0.951086	0.932171	0.951086	0.935611	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
MLPClassifier	0.951429	0.932009	0.951429	0.9347	MiniBatchSparsePCA(n_components=28)
KNeighborsClassifier	0.949943	0.931934	0.949943	0.936647	SelectKBest(f_classif, k=28)
SelfTrainingClassifier	0.949943	0.931934	0.949943	0.936647	SelectKBest(f_classif, k=28)
MLPClassifier	0.951543	0.93149	0.951543	0.933748	PCA(n_components=28)
MLPClassifier	0.9516	0.931321	0.9516	0.93334	RFE(estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.950743	0.931109	0.950743	0.935081	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.950743	0.931109	0.950743	0.935081	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LabelSpreading	0.950743	0.931109	0.950743	0.935081	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelSpreading	0.950743	0.931109	0.950743	0.935081	RFECV(estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.950571	0.930832	0.950571	0.935052	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.950571	0.930832	0.950571	0.935052	RFE(estimator=MLPClassifier(), n_features_to_select=28)
MLPClassifier	0.950571	0.930832	0.950571	0.935052	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelPropagation	0.950571	0.930832	0.950571	0.935052	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelPropagation	0.950571	0.930832	0.950571	0.935052	RFECV(estimator=SVC(), min_features_to_select=28)
PassiveAggressiveClassifier	0.858743	0.930025	0.858743	0.889042	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.951657	0.929623	0.951657	0.931308	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.9512	0.929595	0.9512	0.932834	RFE(estimator=MLPClassifier(), n_features_to_select=28)

SGDClassifier	0.9516	0.92857	0.9516	0.930778	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.951486	0.928435	0.951486	0.931114	SparsePCA(n_components=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	SelectKBest(f_classif, k=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	SelectKBest(mutual_info_classif, k=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	SelectKBest(f_regression, k=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	RFE(estimator=SVC(), n_features_to_select=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	RFECV(estimator=SVC(), min_features_to_select=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.951429	0.928255	0.951429	0.93118	PCA(n_components=28)
SGDClassifier	0.950971	0.927946	0.950971	0.932063	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SGDClassifier	0.951829	0.927942	0.951829	0.929333	RFE(estimator=LogisticRegression(), n_features_to_select=28)
AdaBoostClassifier	0.951429	0.927768	0.951429	0.930884	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.951714	0.92765	0.951714	0.929808	PCA(n_components=28)
SGDClassifier	0.951486	0.926721	0.951486	0.930207	SelectKBest(f_regression, k=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	SelectKBest(f_classif, k=28)

LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	SelectKBest(mutual_info_classif, k=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	SelectKBest(f_regression, k=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	RFE(estimator=SVC(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	RFECV(estimator=SVC(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	PCA(n_components=28)
LinearDiscriminantAnalysis	0.947257	0.92654	0.947257	0.933577	SparsePCA(n_components=28)
AdaBoostClassifier	0.9512	0.926253	0.9512	0.930658	PCA(n_components=28)
PassiveAggressiveClassifier	0.886571	0.926181	0.886571	0.904347	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SGDClassifier	0.9496	0.9259	0.9496	0.932327	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.948743	0.925546	0.948743	0.932568	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.913029	0.925521	0.913029	0.918972	SelectKBest(mutual_info_classif, k=28)
SGDClassifier	0.949657	0.925274	0.949657	0.931928	PCA(n_components=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	SelectKBest(f_classif, k=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	SelectKBest(mutual_info_classif, k=28)

GaussianNB	0.941886	0.92492	0.941886	0.932001	SelectKBest(f_regression, k=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	RFECV(estimator=SVC(), min_features_to_select=28)
GaussianNB	0.941886	0.92492	0.941886	0.932001	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.917371	0.924678	0.917371	0.920914	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.783029	0.924646	0.783029	0.841178	PCA(n_components=28)
PassiveAggressiveClassifier	0.948514	0.924464	0.948514	0.93202	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.925657	0.924425	0.925657	0.925037	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.950571	0.924335	0.950571	0.930702	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926457	0.924294	0.926457	0.925364	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.9264	0.924279	0.9264	0.925328	SelectKBest(f_regression, k=28)
QuadraticDiscriminantAnalysis	0.926457	0.924207	0.926457	0.925319	RFECV(estimator=SVC(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.9264	0.924105	0.9264	0.925239	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.927029	0.924015	0.927029	0.925498	PCA(n_components=28)

QuadraticDiscriminantAnalysis	0.926686	0.924008	0.926686	0.925328	RFE(estimator=MLPClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.9256	0.923978	0.9256	0.924783	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.926686	0.92392	0.926686	0.925284	RFE(estimator=MLPClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.925657	0.923906	0.925657	0.924774	SelectKBest(mutual_info_classif, k=28)
QuadraticDiscriminantAnalysis	0.926286	0.923899	0.926286	0.925078	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926229	0.923884	0.926229	0.925042	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926229	0.923884	0.926229	0.925042	RFE(estimator=LogisticRegression(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.925886	0.923792	0.925886	0.924828	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.782057	0.923763	0.782057	0.840442	SparsePCA(n_components=28)
GaussianNB	0.926	0.923735	0.926	0.924855	SparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.925314	0.92373	0.925314	0.924516	SelectKBest(f_classif, k=28)
GaussianNB	0.927543	0.923718	0.927543	0.925592	PCA(n_components=28)
SGDClassifier	0.950971	0.923689	0.950971	0.930032	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.925486	0.923687	0.925486	0.924578	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.925543	0.923615	0.925543	0.92457	RFE(estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.937371	0.923433	0.937371	0.929697	SelectKBest(f_regression, k=28)
SGDClassifier	0.948914	0.923409	0.948914	0.93132	RFECV(estimator=SVC(), min_features_to_select=28)
GaussianNB	0.934171	0.92329	0.934171	0.928355	MiniBatchSparsePCA(n_components=28)
BernoulliNB	0.8412	0.923129	0.8412	0.876797	MiniBatchSparsePCA(n_components=28)
AdaBoostClassifier	0.951486	0.92181	0.951486	0.929047	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.951543	0.921648	0.951543	0.928968	SelectKBest(f_classif, k=28)
SGDClassifier	0.938057	0.921436	0.938057	0.928823	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SGDClassifier	0.949086	0.921368	0.949086	0.930342	SelectKBest(mutual_info_classif, k=28)
PassiveAggressiveClassifier	0.901714	0.921118	0.901714	0.910906	RFECV(estimator=SVC(), min_features_to_select=28)

SGDClassifier	0.9508	0.919841	0.9508	0.929221	RFE(estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.9468	0.919834	0.9468	0.929927	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SGDClassifier	0.950114	0.919329	0.950114	0.929368	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.946686	0.919312	0.946686	0.929691	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.838743	0.919252	0.838743	0.874354	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.943086	0.919116	0.943086	0.929053	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.941543	0.91736	0.941543	0.927779	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SGDClassifier	0.949714	0.915302	0.949714	0.928547	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SGDClassifier	0.950686	0.914914	0.950686	0.928528	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.951371	0.906169	0.951371	0.92822	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.447314	0.876547	0.447314	0.585029	SelectKBest(f_classif, k=28)

APPENDIX 5: VULNERABILITY FUNCTION LEVEL DATA RESULTS IN 2002 TO 2019

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
RandomForestClassifier	0.950743	0.936812	0.950743	0.938 86	SelectKBest(mutual_info_classif, k=28)
RandomForestClassifier	0.950696	0.936706	0.950696	0.938778	SelectKBest(f_regression, k=28)
RandomForestClassifier	0.950975	0.937009	0.950975	0.938623	PCA(n_components=28)
RandomForestClassifier	0.950556	0.936413	0.950556	0.938579	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.949718	0.935522	0.949718	0.938573	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RandomForestClassifier	0.950649	0.936513	0.950649	0.938545	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RandomForestClassifier	0.950463	0.936258	0.950463	0.938514	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RandomForestClassifier	0.950743	0.936539	0.950743	0.938357	MiniBatchSparsePCA(n_components=28)
RandomForestClassifier	0.950417	0.936089	0.950417	0.938331	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RandomForestClassifier	0.950324	0.935966	0.950324	0.938315	SelectKBest(f_classif, k=28)
RandomForestClassifier	0.950 51	0.936188	0.950 51	0.938295	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RandomForestClassifier	0.950137	0.935629	0.950137	0.938134	RFECV(estimator=SVC(), min_features_to_select=28)
RandomForestClassifier	0.950277	0.935794	0.950277	0.938132	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.950184	0.935672	0.950184	0.938117	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RandomForestClassifier	0.950044	0.935444	0.950044	0.938019	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RandomForestClassifier	0.950696	0.936246	0.950696	0.937912	SparsePCA(n_components=28)
StackingClassifier	0.951068	0.936839	0.951068	0.937856	SparsePCA(n_components=28)
RandomForestClassifier	0.949951	0.935226	0.949951	0.937853	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
BaggingClassifier	0.949 02	0.934304	0.949 02	0.937793	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BaggingClassifier	0.949858	0.935041	0.949858	0.937737	PCA(n_components=28)

StackingClassifier	0.951068	0.936765	0.951068	0.937641	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.949299	0.934331	0.949299	0.937547	RFECV(estimator=SVC(), min_features_to_select=28)
StackingClassifier	0.950836	0.936283	0.950836	0.93748	PCA(n_components=28)
StackingClassifier	0.951022	0.936595	0.951022	0.937392	RFECV(estimator=SVC(), min_features_to_select=28)
BaggingClassifier	0.94888	0.933812	0.94888	0.937354	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BaggingClassifier	0.949532	0.934336	0.949532	0.937256	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.950929	0.936359	0.950929	0.937219	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
StackingClassifier	0.950743	0.936006	0.950743	0.937199	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BaggingClassifier	0.94902	0.933629	0.94902	0.937002	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.950743	0.935901	0.950743	0.936925	SelectKBest(f_classif, k=28)
StackingClassifier	0.950789	0.935982	0.950789	0.936902	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BaggingClassifier	0.948694	0.933242	0.948694	0.936877	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BaggingClassifier	0.948322	0.932912	0.948322	0.936818	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.949392	0.933803	0.949392	0.936746	SparsePCA(n_components=28)
StackingClassifier	0.950882	0.936118	0.950882	0.936743	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.948834	0.933224	0.948834	0.936721	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.94888	0.933206	0.94888	0.936651	SelectKBest(f_classif, k=28)
StackingClassifier	0.95037	0.935071	0.95037	0.936559	SelectKBest(f_regression, k=28)
StackingClassifier	0.950603	0.935495	0.950603	0.936552	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.95051	0.935316	0.95051	0.936544	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.95037	0.934995	0.95037	0.936393	SelectKBest(mutual_info_classif, k=28)
BaggingClassifier	0.948229	0.932407	0.948229	0.936355	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)

StackingClassifier	0.950463	0.935147	0.950463	0.936344	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.947716	0.932025	0.947716	0.936301	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BaggingClassifier	0.947903	0.932066	0.947903	0.936231	SelectKBest(f_regression, k=28)
StackingClassifier	0.950417	0.934999	0.950417	0.9362	RFE(estimator=MLPClassifier(), n_features_to_select=28)
StackingClassifier	0.950184	0.934526	0.950184	0.936098	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BaggingClassifier	0.948042	0.931907	0.948042	0.935973	SelectKBest(mutual_info_classif, k=28)
XGBClassifier	0.949672	0.932097	0.949672	0.933126	SparsePCA(n_components=28)
XGBClassifier	0.949718	0.932175	0.949718	0.933028	PCA(n_components=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	SelectKBest(f_classif, k=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	SelectKBest(mutual_info_classif, k=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	SelectKBest(f_regression, k=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	RFE(estimator=LogisticRegression(), n_features_to_select=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	RFE(estimator=SVC(), n_features_to_select=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	RFE(estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	RFECV(estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.949765	0.932104	0.949765	0.932472	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
XGBClassifier	0.948927	0.929978	0.948927	0.932393	MiniBatchSparsePCA(n_components=28)

KNeighborsClassifier	0.948601	0.92924	0.948601	0.932249	SparsePCA(n_components=28)
SelfTrainingClassifier	0.948601	0.92924	0.948601	0.932249	SparsePCA(n_components=28)
ExtraTreeClassifier	0.934261	0.928801	0.934261	0.931385	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.949718	0.931637	0.949718	0.931288	SparsePCA(n_components=28)
ExtraTreeClassifier	0.934448	0.928443	0.934448	0.931269	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.947996	0.927121	0.947996	0.931096	MiniBatchSparsePCA(n_components=28)
SelfTrainingClassifier	0.947996	0.927121	0.947996	0.931096	MiniBatchSparsePCA(n_components=28)
ExtraTreeClassifier	0.933889	0.928267	0.933889	0.930927	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.933377	0.928052	0.933377	0.930583	RFE(estimator=MLPClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.933144	0.92783	0.933144	0.930358	RFE(estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.932399	0.928411	0.932399	0.930334	SelectKBest(f_regression, k=28)
ExtraTreeClassifier	0.933004	0.927645	0.933004	0.930195	RFE(estimator=SVC(), n_features_to_select=28)
ExtraTreeClassifier	0.933144	0.927448	0.933144	0.930149	SelectKBest(mutual_info_classif, k=28)
KNeighborsClassifier	0.945761	0.923561	0.945761	0.930147	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.945761	0.923561	0.945761	0.930147	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.945761	0.923561	0.945761	0.930147	RFE(estimator=SVC(), n_features_to_select=28)
SelfTrainingClassifier	0.945761	0.923561	0.945761	0.930147	RFE(estimator=SVC(), n_features_to_select=28)
KNeighborsClassifier	0.945761	0.923561	0.945761	0.930147	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.945761	0.923561	0.945761	0.930147	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.945761	0.923561	0.945761	0.930147	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SelfTrainingClassifier	0.945761	0.923561	0.945761	0.930147	RFECV(estimator=LogisticRegression(), min_features_to_select=28)

KNeighborsClassifier	0.945761	0.923561	0.945761	0.930147	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.945761	0.923561	0.945761	0.930147	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.945854	0.923613	0.945854	0.930143	SelectKBest(f_classif, k=28)
SelfTrainingClassifier	0.945854	0.923613	0.945854	0.930143	SelectKBest(f_classif, k=28)
KNeighborsClassifier	0.945854	0.923613	0.945854	0.930143	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.945854	0.923613	0.945854	0.930143	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelSpreading	0.949346	0.929848	0.949346	0.930137	SparsePCA(n_components=28)
KNeighborsClassifier	0.945808	0.923542	0.945808	0.930114	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SelfTrainingClassifier	0.945808	0.923542	0.945808	0.930114	RFE(estimator=LogisticRegression(), n_features_to_select=28)
KNeighborsClassifier	0.945808	0.923542	0.945808	0.930114	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.945808	0.923542	0.945808	0.930114	RFE(estimator=MLPClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.945808	0.923542	0.945808	0.930114	RFECV(estimator=SVC(), min_features_to_select=28)
SelfTrainingClassifier	0.945808	0.923542	0.945808	0.930114	RFECV(estimator=SVC(), min_features_to_select=28)
KNeighborsClassifier	0.945761	0.923472	0.945761	0.930085	SelectKBest(f_regression, k=28)
SelfTrainingClassifier	0.945761	0.923472	0.945761	0.930085	SelectKBest(f_regression, k=28)
KNeighborsClassifier	0.945761	0.923472	0.945761	0.930085	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.945761	0.923472	0.945761	0.930085	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.945621	0.923354	0.945621	0.930061	SelectKBest(mutual_info_classif, k=28)
SelfTrainingClassifier	0.945621	0.923354	0.945621	0.930061	SelectKBest(mutual_info_classif, k=28)
LGBMClassifier	0.950417	0.936409	0.950417	0.929957	SparsePCA(n_components=28)
ExtraTreeClassifier	0.933097	0.927108	0.933097	0.929942	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.944923	0.922474	0.944923	0.929688	PCA(n_components=28)

SelfTrainingClassifier	0.944923	0.922474	0.944923	0.929688	PCA(n_components=28)
LabelPropagation	0.948415	0.926331	0.948415	0.929506	MiniBatchSparsePCA(n_components=28)
LGBMClassifier	0.950137	0.93466	0.950137	0.929485	PCA(n_components=28)
HistGradientBoostingClassifier	0.949998	0.933602	0.949998	0.929403	SparsePCA(n_components=28)
ExtraTreeClassifier	0.932166	0.926794	0.932166	0.929357	PCA(n_components=28)
ExtraTreeClassifier	0.932539	0.926425	0.932539	0.929321	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.93184	0.926924	0.93184	0.92928	RFECV(estimator=SVC(), min_features_to_select=28)
ExtraTreeClassifier	0.932213	0.926489	0.932213	0.929212	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	SelectKBest(f_classif, k=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	SelectKBest(mutual_info_classif, k=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	SelectKBest(f_regression, k=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	RFE(estimator=SVC(), n_features_to_select=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	RFECV(estimator=SVC(), min_features_to_select=28)
LGBMClassifier	0.949672	0.931406	0.949672	0.929136	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.932585	0.926052	0.932585	0.929136	RFECV(estimator=SVC(), min_features_to_select=28)

MLPClassifier	0.948973	0.927674	0.948973	0.928956	SelectKBest(f_classif, k=28)
DecisionTreeClassifier	0.931468	0.926521	0.931468	0.928893	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
MLPClassifier	0.948229	0.925136	0.948229	0.928 89	SelectKBest(f_regression, k=28)
LabelSpreading	0.948368	0.925442	0.948368	0.928825	MiniBatchSparsePCA(n_components=28)
LabelPropagation	0.947111	0.922841	0.947111	0.928 8	SelectKBest(f_classif, k=28)
LabelPropagation	0.947111	0.922841	0.947111	0.928 8	SelectKBest(f_regression, k=28)
LabelPropagation	0.947111	0.922841	0.947111	0.928 8	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelPropagation	0.947111	0.922841	0.947111	0.928 8	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.947111	0.922841	0.947111	0.928 8	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LabelPropagation	0.947111	0.922841	0.947111	0.928 8	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelPropagation	0.947111	0.922841	0.947111	0.928 8	RFECV(estimator=SVC(), min_features_to_select=28)
MLPClassifier	0.948415	0.925523	0.948415	0.928779	SparsePCA(n_components=28)
MLPClassifier	0.947996	0.924383	0.947996	0.928753	SelectKBest(mutual_info_classif, k=28)
DecisionTreeClassifier	0.931328	0.926337	0.931328	0.928731	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.946785	0.922225	0.946785	0.928675	SelectKBest(mutual_info_classif, k=28)
LabelPropagation	0.946785	0.922225	0.946785	0.928675	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.947484	0.923223	0.947484	0.928667	SelectKBest(f_classif, k=28)
LabelSpreading	0.947484	0.923223	0.947484	0.928667	SelectKBest(f_regression, k=28)
LabelSpreading	0.947484	0.923223	0.947484	0.928667	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelSpreading	0.947484	0.923223	0.947484	0.928667	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.947484	0.923223	0.947484	0.928667	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LabelSpreading	0.947484	0.923223	0.947484	0.928667	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

LabelSpreading	0.947484	0.923223	0.947484	0.928667	RFECV(estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.946739	0.922126	0.946739	0.928647	RFE(estimator=SVC(), n_features_to_select=28)
LabelPropagation	0.946739	0.922126	0.946739	0.928647	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelPropagation	0.946739	0.922126	0.946739	0.928647	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.946739	0.922126	0.946739	0.928647	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.931282	0.926189	0.931282	0.92863	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
DecisionTreeClassifier	0.931142	0.926266	0.931142	0.928607	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
MLPClassifier	0.948461	0.925436	0.948461	0.928583	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.947158	0.92251	0.947158	0.928546	SelectKBest(mutual_info_classif, k=28)
LabelSpreading	0.947158	0.92251	0.947158	0.928546	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.947111	0.922398	0.947111	0.928518	RFE(estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.947111	0.922398	0.947111	0.928518	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelSpreading	0.947111	0.922398	0.947111	0.928518	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.947111	0.922398	0.947111	0.928518	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelPropagation	0.946413	0.921459	0.946413	0.928454	PCA(n_components=28)
MLPClassifier	0.948461	0.925255	0.948461	0.928432	RFE(estimator=MLPClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.949486	0.930277	0.949486	0.928406	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.930723	0.926237	0.930723	0.9284	SparsePCA(n_components=28)
DecisionTreeClassifier	0.930863	0.926029	0.930863	0.928353	SelectKBest(mutual_info_classif, k=28)
DecisionTreeClassifier	0.931095	0.92579	0.931095	0.92833	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelSpreading	0.946785	0.921648	0.946785	0.928326	PCA(n_components=28)

HistGradientBoostingClassifier	0.949206	0.928406	0.949206	0.928247	SelectKBest(f_classif, k=28)
HistGradientBoostingClassifier	0.949067	0.927613	0.949067	0.928245	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
MLPClassifier	0.947251	0.922158	0.947251	0.928243	PCA(n_components=28)
MLPClassifier	0.948508	0.925166	0.948508	0.928232	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
HistGradientBoostingClassifier	0.949532	0.930691	0.949532	0.928196	SelectKBest(mutual_info_classif, k=28)
HistGradientBoostingClassifier	0.949905	0.934473	0.949905	0.928165	PCA(n_components=28)
ExtraTreeClassifier	0.931421	0.925122	0.931421	0.928114	SparsePCA(n_components=28)
DecisionTreeClassifier	0.930397	0.925 92	0.930397	0.928 08	SelectKBest(f_classif, k=28)
MLPClassifier	0.948694	0.925644	0.948694	0.928032	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.930304	0.925886	0.930304	0.928019	SelectKBest(f_classif, k=28)
HistGradientBoostingClassifier	0.949206	0.928306	0.949206	0.928011	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.949253	0.928598	0.949253	0.927958	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.949206	0.928274	0.949206	0.927931	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.948787	0.925942	0.948787	0.927 93	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.929885	0.926058	0.929885	0.927915	PCA(n_components=28)
DecisionTreeClassifier	0.930397	0.925593	0.930397	0.927906	SelectKBest(f_regression, k=28)
LinearDiscriminantAnalysis	0.943154	0.919152	0.943154	0.927866	MiniBatchSparsePCA(n_components=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	SelectKBest(f_classif, k=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	SelectKBest(mutual_info_classif, k=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	SelectKBest(f_regression, k=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	RFE(estimator=LogisticRegression(), n_features_to_select=28)

LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	RFE(estimator=SVC(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	RFECV(estimator=SVC(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	PCA(n_components=28)
LinearDiscriminantAnalysis	0.943107	0.919102	0.943107	0.927838	SparsePCA(n_components=28)
MLPClassifier	0.948834	0.926027	0.948834	0.927799	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.930025	0.925587	0.930025	0.927731	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
MLPClassifier	0.949113	0.927554	0.949113	0.927719	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.929978	0.925438	0.929978	0.92763	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.949067	0.927182	0.949067	0.927613	RFECV(estimator=SVC(), min_features_to_select=28)
HistGradientBoostingClassifier	0.949346	0.929262	0.949346	0.927609	RFE(estimator=LogisticRegression(), n_features_to_select=28)
DecisionTreeClassifier	0.929838	0.925519	0.929838	0.927608	RFE(estimator=MLPClassifier(), n_features_to_select=28)
MLPClassifier	0.94902	0.926866	0.94902	0.927587	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.929606	0.9255	0.929606	0.92749	RFE(estimator=SVC(), n_features_to_select=28)
MLPClassifier	0.948694	0.924968	0.948694	0.927484	RFECV(estimator=SVC(), min_features_to_select=28)
DecisionTreeClassifier	0.929559	0.925418	0.929559	0.927425	MiniBatchSparsePCA(n_components=28)
DecisionTreeClassifier	0.929699	0.925205	0.929699	0.927376	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.94902	0.926667	0.94902	0.927346	RFE(estimator=SVC(), n_features_to_select=28)

PassiveAggressiveClassifier	0.940081	0.918775	0.940081	0.927335	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.948601	0.923839	0.948601	0.927032	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
MLPClassifier	0.94888	0.925406	0.94888	0.927025	RFE(estimator=SVC(), n_features_to_select=28)
HistGradientBoostingClassifier	0.949299	0.928872	0.949299	0.927009	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SGDClassifier	0.946832	0.918759	0.946832	0.926821	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.94902	0.926139	0.94902	0.926773	SelectKBest(f_regression, k=28)
HistGradientBoostingClassifier	0.948927	0.925344	0.948927	0.926722	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
VotingClassifier	0.948694	0.923565	0.948694	0.926595	PCA(n_components=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	SelectKBest(f_classif, k=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	SelectKBest(mutual_info_classif, k=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	SelectKBest(f_regression, k=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFE(estimator=LogisticRegression(), n_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFE(estimator=SVC(), n_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFE(estimator=MLPClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFECV(estimator=SVC(), min_features_to_select=28)
AdaBoostClassifier	0.949113	0.926818	0.949113	0.926574	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
VotingClassifier	0.948741	0.923747	0.948741	0.926538	SparsePCA(n_components=28)

MLPClassifier	0.948322	0.921418	0.948322	0.926473	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.945389	0.916767	0.945389	0.926444	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.948834	0.924182	0.948834	0.926422	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.948834	0.924182	0.948834	0.926422	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.949439	0.931409	0.949439	0.92641	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.949439	0.931409	0.949439	0.92641	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.949439	0.931409	0.949439	0.92641	RFE(estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.949439	0.931409	0.949439	0.92641	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.949439	0.931409	0.949439	0.92641	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.949439	0.931409	0.949439	0.92641	RFECV(estimator=SVC(), min_features_to_select=28)
VotingClassifier	0.948787	0.923795	0.948787	0.926397	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.949392	0.930606	0.949392	0.926385	SelectKBest(f_regression, k=28)
GradientBoostingClassifier	0.949392	0.930606	0.949392	0.926385	RFE(estimator=LogisticRegression(), n_features_to_select=28)
VotingClassifier	0.948741	0.923421	0.948741	0.926372	SelectKBest(f_regression, k=28)
VotingClassifier	0.948741	0.923421	0.948741	0.926372	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
VotingClassifier	0.948694	0.92306	0.948694	0.926347	RFE(estimator=SVC(), n_features_to_select=28)
VotingClassifier	0.948694	0.92306	0.948694	0.926347	RFE(estimator=MLPClassifier(), n_features_to_select=28)
VotingClassifier	0.948694	0.92306	0.948694	0.926347	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
VotingClassifier	0.948834	0.924028	0.948834	0.926339	RFECV(estimator=SVC(), min_features_to_select=28)
GradientBoostingClassifier	0.949392	0.930762	0.949392	0.9263	SelectKBest(f_classif, k=28)
GradientBoostingClassifier	0.949392	0.930762	0.949392	0.9263	SelectKBest(mutual_info_classif, k=28)

GradientBoostingClassifier	0.949392	0.930762	0.949392	0.9263	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.949346	0.929953	0.949346	0.926275	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.949346	0.929953	0.949346	0.926275	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
VotingClassifier	0.948694	0.922874	0.948694	0.926263	SelectKBest(f_classif, k=28)
VotingClassifier	0.948648	0.922515	0.948648	0.926238	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
VotingClassifier	0.948648	0.922515	0.948648	0.926238	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.949113	0.926497	0.949113	0.92615	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.948927	0.924444	0.948927	0.926136	SparsePCA(n_components=28)
VotingClassifier	0.948601	0.921953	0.948601	0.926129	SelectKBest(mutual_info_classif, k=28)
VotingClassifier	0.948694	0.922249	0.948694	0.926011	MiniBatchSparsePCA(n_components=28)
AdaBoostClassifier	0.949113	0.926351	0.949113	0.925979	MiniBatchSparsePCA(n_components=28)
GradientBoostingClassifier	0.949253	0.928593	0.949253	0.925967	SparsePCA(n_components=28)
GradientBoostingClassifier	0.949299	0.929597	0.949299	0.925905	PCA(n_components=28)
LogisticRegression	0.948554	0.920856	0.948554	0.925852	SparsePCA(n_components=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	SelectKBest(f_classif, k=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	SelectKBest(mutual_info_classif, k=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	SelectKBest(f_regression, k=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	RFE(estimator=MLPClassifier(), n_features_to_select=28)

GaussianNB	0.936915	0.917399	0.936915	0.925769	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	RFECV(estimator=SVC(), min_features_to_select=28)
GaussianNB	0.936915	0.917399	0.936915	0.925769	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.948554	0.920571	0.948554	0.925767	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.946133	0.915112	0.946133	0.925667	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.948834	0.922382	0.948834	0.925659	SparsePCA(n_components=28)
SGDClassifier	0.944597	0.914653	0.944597	0.925634	SelectKBest(mutual_info_classif, k=28)
PassiveAggressiveClassifier	0.9432	0.914724	0.9432	0.925616	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.949113	0.925897	0.949113	0.925545	MiniBatchSparsePCA(n_components=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	SelectKBest(f_classif, k=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	SelectKBest(mutual_info_classif, k=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	SelectKBest(f_regression, k=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	RFE(estimator=SVC(), n_features_to_select=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

LogisticRegression	0.948415	0.91883	0.948415	0.925523	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	RFECV(estimator=SVC(), min_features_to_select=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.948415	0.91883	0.948415	0.925523	PCA(n_components=28)
SVC	0.949206	0.927909	0.949206	0.925506	SelectKBest(f_classif, k=28)
SVC	0.949206	0.927909	0.949206	0.925506	SelectKBest(mutual_info_classif, k=28)
SVC	0.949206	0.927909	0.949206	0.925506	SelectKBest(f_regression, k=28)
SVC	0.949206	0.927909	0.949206	0.925506	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SVC	0.949206	0.927909	0.949206	0.925506	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SVC	0.949206	0.927909	0.949206	0.925506	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SVC	0.949206	0.927909	0.949206	0.925506	RFE(estimator=SVC(), n_features_to_select=28)
SVC	0.949206	0.927909	0.949206	0.925506	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SVC	0.949206	0.927909	0.949206	0.925506	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.949206	0.927909	0.949206	0.925506	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SVC	0.949206	0.927909	0.949206	0.925506	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SVC	0.949206	0.927909	0.949206	0.925506	RFECV(estimator=SVC(), min_features_to_select=28)
SVC	0.949206	0.927909	0.949206	0.925506	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.949206	0.927909	0.949206	0.925506	PCA(n_components=28)
SGDClassifier	0.948508	0.918801	0.948508	0.925401	PCA(n_components=28)
SGDClassifier	0.948368	0.917688	0.948368	0.925328	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.941524	0.914523	0.941524	0.925317	RFECV(estimator=SVC(), min_features_to_select=28)
SVC	0.948834	0.921045	0.948834	0.925312	MiniBatchSparsePCA(n_components=28)

RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(f_classif, k=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(f_classif, k=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(mutual_info_classif, k=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(mutual_info_classif, k=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(f_regression, k=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	SelectKBest(f_regression, k=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=LogisticRegression(), min_features_to_select=28)

RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	PCA(n_components=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	PCA(n_components=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	SparsePCA(n_components=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	SparsePCA(n_components=28)
RidgeClassifier	0.949 16	0.926 722	0.949 16	0.925 305	MiniBatchSparsePCA(n_components=28)
RidgeClassifierCV	0.949 16	0.926 722	0.949 16	0.925 305	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.937 986	0.915 961	0.937 986	0.925 278	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.948 508	0.917 913	0.948 508	0.925 229	PCA(n_components=28)
PassiveAggressiveClassifier	0.943 992	0.913 656	0.943 992	0.925 221	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SGDClassifier	0.946 832	0.913 567	0.946 832	0.925 097	RFE(estimator=SVC(), n_features_to_select=28)
SGDClassifier	0.948 554	0.916 679	0.948 554	0.924 992	SelectKBest(f_regression, k=28)
SGDClassifier	0.948 415	0.915 388	0.948 415	0.924 92	RFE(estimator=LogisticRegression(), n_features_to_select=28)
PassiveAggressiveClassifier	0.948 834	0.918 113	0.948 834	0.924 871	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SGDClassifier	0.949 113	0.924 21	0.949 113	0.924 834	RFECV(estimator=SVC(), min_features_to_select=28)
SGDClassifier	0.948 694	0.914 986	0.948 694	0.924 71	SelectKBest(f_classif, k=28)
SGDClassifier	0.949 253	0.951 828	0.949 253	0.924 633	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SGDClassifier	0.948 508	0.912 843	0.948 508	0.924 615	RFE(estimator=MLPClassifier(), n_features_to_select=28)

SGDClassifier	0.948648	0.913183	0.948648	0.924597	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SGDClassifier	0.949067	0.917926	0.949067	0.924539	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SGDClassifier	0.948089	0.910749	0.948089	0.924489	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
PassiveAggressiveClassifier	0.946971	0.91037	0.946971	0.924426	RFE(estimator=LogisticRegression(), n_features_to_select=28)
PassiveAggressiveClassifier	0.947344	0.909202	0.947344	0.92428	PCA(n_components=28)
PassiveAggressiveClassifier	0.940407	0.911924	0.940407	0.92389	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
PassiveAggressiveClassifier	0.937986	0.912549	0.937986	0.92352	RFE(estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.931421	0.915563	0.931421	0.922825	SelectKBest(mutual_info_classif, k=28)
PassiveAggressiveClassifier	0.944597	0.90414	0.944597	0.9227	SelectKBest(f_regression, k=28)
PassiveAggressiveClassifier	0.93915	0.909569	0.93915	0.922519	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.940267	0.906113	0.940267	0.921594	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GaussianNB	0.925741	0.916848	0.925741	0.921106	PCA(n_components=28)
QuadraticDiscriminantAnalysis	0.925788	0.916627	0.925788	0.921009	PCA(n_components=28)
QuadraticDiscriminantAnalysis	0.924903	0.917216	0.924903	0.920919	MiniBatchSparsePCA(n_components=28)
GaussianNB	0.92495	0.916767	0.92495	0.920702	SparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.92495	0.91614	0.92495	0.92037	SelectKBest(f_regression, k=28)
QuadraticDiscriminantAnalysis	0.924531	0.916489	0.924531	0.920363	RFECV(estimator=SVC(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924205	0.916628	0.924205	0.920286	RFE(estimator=MLPClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924065	0.916588	0.924065	0.9202	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.924019	0.916575	0.924019	0.920172	SparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.924112	0.916369	0.924112	0.920106	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.923553	0.916597	0.923553	0.919968	RFE(estimator=LogisticRegression(), n_features_to_select=28)

QuadraticDiscriminantAnalysis	0.9236	0.916533	0.9236	0.919956	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.923414	0.916635	0.923414	0.919922	SelectKBest(mutual_info_classif, k=28)
QuadraticDiscriminantAnalysis	0.923274	0.916596	0.923274	0.919837	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.923134	0.916711	0.923134	0.919831	SelectKBest(f_classif, k=28)
QuadraticDiscriminantAnalysis	0.923227	0.916584	0.923227	0.919808	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.923274	0.91652	0.923274	0.919796	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.923227	0.916507	0.923227	0.919768	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.923553	0.915428	0.923553	0.919352	MiniBatchSparsePCA(n_components=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	SelectKBest(f_classif, k=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	SelectKBest(mutual_info_classif, k=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	SelectKBest(f_regression, k=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	RFE(estimator=LogisticRegression(), n_features_to_select=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	RFE(estimator=SVC(), n_features_to_select=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	RFE(estimator=MLPClassifier(), n_features_to_select=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	RFECV(estimator=SVC(), min_features_to_select=28)
CategoricalNB	0.916104	0.920773	0.916104	0.918391	RFECV(estimator=MLPClassifier(), min_features_to_select=28)

PassiveAggressiveClassifier	0.830392	0.917257	0.830392	0.86812	SelectKBest(f_classif, k=28)
PassiveAggressiveClassifier	0.828344	0.912276	0.828344	0.865637	MiniBatchSparsePCA(n_components=28)
BernoulliNB	0.820802	0.916849	0.820802	0.862238	SparsePCA(n_components=28)
BernoulliNB	0.823735	0.908522	0.823735	0.861923	MiniBatchSparsePCA(n_components=28)
BernoulliNB	0.819871	0.917136	0.819871	0.861737	PCA(n_components=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	SelectKBest(f_classif, k=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	SelectKBest(mutual_info_classif, k=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	SelectKBest(f_regression, k=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	RFECV(estimator=SVC(), min_features_to_select=28)
BernoulliNB	0.764375	0.927738	0.764375	0.828126	RFECV(estimator=MLPClassifier(), min_features_to_select=28)

APPENDIX 6: VULNERABILITY CLASS LEVEL DATA RESULTS

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
RandomForestClassifier	0.947347	0.936937	0.947347	0.940376	RFECV(estimator=SVC(), min_features_to_select=28)
BaggingClassifier	0.946722	0.936175	0.946722	0.939751	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BaggingClassifier	0.947138	0.936016	0.947138	0.939561	SelectKBest(mutual_info_classif, k=28)
RandomForestClassifier	0.946722	0.935988	0.946722	0.939587	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.946514	0.935986	0.946514	0.939597	RFECV(estimator=SVC(), min_features_to_select=28)
RandomForestClassifier	0.946514	0.935796	0.946514	0.939434	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RandomForestClassifier	0.946722	0.935614	0.946722	0.939254	SelectKBest(f_regression, k=28)
RandomForestClassifier	0.946722	0.935427	0.946722	0.939085	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.946098	0.935225	0.946098	0.938962	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RandomForestClassifier	0.946306	0.935223	0.946306	0.938948	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RandomForestClassifier	0.945473	0.935075	0.945473	0.938832	RFE(estimator=SVC(), n_features_to_select=28)
RandomForestClassifier	0.947138	0.934936	0.947138	0.938516	MiniBatchSparsePCA(n_components=28)
LGBMClassifier	0.946722	0.934868	0.946722	0.938567	PCA(n_components=28)
RandomForestClassifier	0.945682	0.934854	0.945682	0.938657	SelectKBest(mutual_info_classif, k=28)
RandomForestClassifier	0.94589	0.934842	0.94589	0.938643	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BaggingClassifier	0.946098	0.934836	0.946098	0.938627	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
XGBClassifier	0.946514	0.93466	0.946514	0.938415	MiniBatchSparsePCA(n_components=28)
XGBClassifier	0.946306	0.934646	0.946306	0.938437	PCA(n_components=28)
BaggingClassifier	0.946722	0.934496	0.946722	0.938212	MiniBatchSparsePCA(n_components=28)
RandomForestClassifier	0.945682	0.934255	0.945682	0.938153	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.946306	0.93407	0.946306	0.937909	SparsePCA(n_components=28)

RandomForestClassifier	0.945682	0.934055	0.945682	0.937982	SelectKBest(f_classif, k=28)
BaggingClassifier	0.94589	0.934052	0.94589	0.93796	SelectKBest(f_regression, k=28)
LGBMClassifier	0.947138	0.933692	0.947138	0.937208	KernelPCA(kernel="linear", n_components=28)
RandomForestClassifier	0.946306	0.933685	0.946306	0.937548	KernelPCA(kernel="linear", n_components=28)
BaggingClassifier	0.945682	0.933654	0.945682	0.937633	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BaggingClassifier	0.945682	0.933252	0.945682	0.937277	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.945057	0.933072	0.945057	0.937182	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.945265	0.933057	0.945265	0.937155	KernelPCA(kernel="linear", n_components=28)
DecisionTreeClassifier	0.937357	0.932999	0.937357	0.935044	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LGBMClassifier	0.946306	0.932916	0.946306	0.936801	MiniBatchSparsePCA(n_components=28)
RandomForestClassifier	0.945265	0.93285	0.945265	0.936977	PCA(n_components=28)
HistGradientBoostingClassifier	0.946098	0.932687	0.946098	0.936651	SparsePCA(n_components=28)
BaggingClassifier	0.945682	0.932646	0.945682	0.936729	PCA(n_components=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	SelectKBest(f_classif, k=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	SelectKBest(mutual_info_classif, k=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	SelectKBest(f_regression, k=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.948179	0.932614	0.948179	0.928172	PCA(n_components=28)
RandomForestClassifier	0.945057	0.932233	0.945057	0.936465	SparsePCA(n_components=28)
RandomForestClassifier	0.944017	0.932153	0.944017	0.936433	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.947138	0.932128	0.947138	0.935367	KernelPCA(kernel="linear", n_components=28)
ExtraTreeClassifier	0.938398	0.932093	0.938398	0.934948	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BaggingClassifier	0.944641	0.932053	0.944641	0.936349	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
DecisionTreeClassifier	0.934443	0.93187	0.934443	0.933117	SelectKBest(f_classif, k=28)
XGBClassifier	0.944433	0.931862	0.944433	0.9362	SparsePCA(n_components=28)
DecisionTreeClassifier	0.934651	0.931714	0.934651	0.93313	RFE(estimator=SVC(), n_features_to_select=28)
StackingClassifier	0.94693	0.931656	0.94693	0.935004	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.946514	0.931638	0.946514	0.935351	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
ExtraTreeClassifier	0.936941	0.93157	0.936941	0.934058	SelectKBest(f_regression, k=28)
ExtraTreeClassifier	0.939438	0.931474	0.939438	0.93495	RFE(estimator=SVC(), n_features_to_select=28)
HistGradientBoostingClassifier	0.946514	0.931449	0.946514	0.935141	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BaggingClassifier	0.9436	0.931351	0.9436	0.935784	RFE(estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.944225	0.931231	0.944225	0.935687	KernelPCA(kernel="linear", n_components=28)
PassiveAggressiveClassifier	0.920708	0.931183	0.920708	0.925552	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.934086	0.931059	0.934086	0.932871	SelectKBest(mutual_info_classif, k=28)
BaggingClassifier	0.944017	0.93104	0.944017	0.935539	SelectKBest(f_classif, k=28)
DecisionTreeClassifier	0.934651	0.930963	0.934651	0.932725	RFE(estimator=LogisticRegression(), n_features_to_select=28)

BernoulliNB	0.867222	0.930958	0.867222	0.893352	SelectKBest(f_classif, k=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	SelectKBest(mutual_info_classif, k=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	SelectKBest(f_regression, k=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	RFECV(estimator=SVC(), min_features_to_select=28)
BernoulliNB	0.867222	0.930958	0.867222	0.893352	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.946993	0.930952	0.946993	0.934113	KernelPCA(kernel="linear", n_components=28)
MLPClassifier	0.946514	0.930882	0.946514	0.934494	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.936524	0.930862	0.936524	0.933348	SparsePCA(n_components=28)
StackingClassifier	0.946306	0.930788	0.946306	0.934567	SelectKBest(f_regression, k=28)
DecisionTreeClassifier	0.934651	0.930711	0.934651	0.932588	RFE(estimator=MLPClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.935692	0.930699	0.935692	0.933037	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.946722	0.930629	0.946722	0.933969	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.945473	0.930562	0.945473	0.934825	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)

ExtraTreeClassifier	0.937357	0.930545	0.937357	0.933628	RFE(estimator=MLPClassifier(), n_features_to_select=28)
StackingClassifier	0.946514	0.930503	0.946514	0.934051	RFECV(estimator=SVC(), min_features_to_select=28)
ExtraTreeClassifier	0.934027	0.930427	0.934027	0.932151	SelectKBest(mutual_info_classif, k=28)
DecisionTreeClassifier	0.934027	0.930427	0.934027	0.932151	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.942768	0.930426	0.942768	0.935015	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
StackingClassifier	0.946306	0.930397	0.946306	0.934129	SelectKBest(mutual_info_classif, k=28)
DecisionTreeClassifier	0.933819	0.930333	0.933819	0.932006	SelectKBest(f_regression, k=28)
DecisionTreeClassifier	0.933819	0.930333	0.933819	0.932006	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.933819	0.930333	0.933819	0.932006	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	SelectKBest(f_classif, k=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	SelectKBest(mutual_info_classif, k=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	SelectKBest(f_regression, k=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	RFE(estimator=SVC(), n_features_to_select=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LGBMClassifier	0.945265	0.930322	0.945265	0.934679	RFECV(estimator=SVC(), min_features_to_select=28)

LGBMClassifier	0.945265	0.930322	0.945265	0.934679	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.946098	0.930309	0.946098	0.934204	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.9436	0.930206	0.9436	0.934874	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
StackingClassifier	0.946306	0.930202	0.946306	0.933907	PCA(n_components=28)
DecisionTreeClassifier	0.934027	0.930172	0.934027	0.932013	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.933403	0.930149	0.933403	0.931716	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.934443	0.930107	0.934443	0.932164	KernelPCA(kernel="linear", n_components=28)
BaggingClassifier	0.943809	0.929934	0.943809	0.934643	SparsePCA(n_components=28)
VotingClassifier	0.94693	0.929911	0.94693	0.932694	KernelPCA(kernel="linear", n_components=28)
MLPClassifier	0.946098	0.929906	0.946098	0.933763	RFE(estimator=SVC(), n_features_to_select=28)
ExtraTreeClassifier	0.9359	0.929789	0.9359	0.932607	RFECV(estimator=SVC(), min_features_to_select=28)
StackingClassifier	0.946098	0.929704	0.946098	0.933538	RFE(estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.935692	0.929685	0.935692	0.932461	SelectKBest(f_classif, k=28)
DecisionTreeClassifier	0.934027	0.929659	0.934027	0.931733	RFECV(estimator=SVC(), min_features_to_select=28)
PassiveAggressiveClassifier	0.912175	0.929585	0.912175	0.920028	PCA(n_components=28)
ExtraTreeClassifier	0.934235	0.929498	0.934235	0.931736	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
StackingClassifier	0.945473	0.929493	0.945473	0.93377	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.947555	0.929471	0.947555	0.928417	KernelPCA(kernel="linear", n_components=28)
RidgeClassifierCV	0.947555	0.929471	0.947555	0.928417	KernelPCA(kernel="linear", n_components=28)
RidgeClassifier	0.947555	0.929471	0.947555	0.928417	SparsePCA(n_components=28)
RidgeClassifierCV	0.947555	0.929471	0.947555	0.928417	SparsePCA(n_components=28)
RidgeClassifier	0.947555	0.929471	0.947555	0.928417	MiniBatchSparsePCA(n_components=28)
RidgeClassifierCV	0.947555	0.929471	0.947555	0.928417	MiniBatchSparsePCA(n_components=28)

ExtraTreeClassifier	0.936733	0.929453	0.936733	0.932741	PCA(n_components=28)
DecisionTreeClassifier	0.935068	0.929377	0.935068	0.932026	MiniBatchSparsePCA(n_components=28)
StackingClassifier	0.94589	0.9292	0.94589	0.933167	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.935692	0.929171	0.935692	0.932165	RFE(estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.935484	0.929066	0.935484	0.932021	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
StackingClassifier	0.945473	0.929057	0.945473	0.933331	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
MLPClassifier	0.9436	0.929029	0.9436	0.933915	SelectKBest(f_regression, k=28)
DecisionTreeClassifier	0.932986	0.928933	0.932986	0.93087	PCA(n_components=28)
ExtraTreeClassifier	0.932986	0.928933	0.932986	0.93087	SparsePCA(n_components=28)
StackingClassifier	0.945682	0.928905	0.945682	0.933024	RFE(estimator=MLPClassifier(), n_features_to_select=28)
StackingClassifier	0.945473	0.928837	0.945473	0.933108	SelectKBest(f_classif, k=28)
StackingClassifier	0.945473	0.928837	0.945473	0.933108	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
VotingClassifier	0.946306	0.928817	0.946306	0.932271	SparsePCA(n_components=28)
SGDClassifier	0.927992	0.928784	0.927992	0.928385	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
MLPClassifier	0.94589	0.928777	0.94589	0.932705	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
ExtraTreeClassifier	0.931738	0.928658	0.931738	0.930149	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
MLPClassifier	0.943184	0.928632	0.943184	0.933624	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	SelectKBest(f_classif, k=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	SelectKBest(mutual_info_classif, k=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	SelectKBest(f_regression, k=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)

RidgeClassifier	0.947347	0.928566	0.947347	0.927662	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifier	0.947347	0.928566	0.947347	0.927662	PCA(n_components=28)
GradientBoostingClassifier	0.94589	0.928564	0.94589	0.93247	KernelPCA(kernel="linear", n_components=28)
AdaBoostClassifier	0.94589	0.928564	0.94589	0.93247	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.945057	0.92852	0.945057	0.933045	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.944849	0.928492	0.944849	0.933122	SelectKBest(f_regression, k=28)
MLPClassifier	0.946098	0.928471	0.946098	0.932131	SelectKBest(mutual_info_classif, k=28)
PassiveAggressiveClassifier	0.946514	0.928405	0.946514	0.93141	SelectKBest(f_regression, k=28)
MLPClassifier	0.945473	0.928393	0.945473	0.932653	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.947138	0.928367	0.947138	0.929356	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.925078	0.928318	0.925078	0.926656	RFE(estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.93153	0.928307	0.93153	0.929866	KernelPCA(kernel="linear", n_components=28)
GradientBoostingClassifier	0.945682	0.92825	0.945682	0.932328	SparsePCA(n_components=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	SelectKBest(f_classif, k=28)

XGBClassifier	0.942352	0.928127	0.942352	0.93324	SelectKBest(mutual_info_classif, k=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	SelectKBest(f_regression, k=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	RFE(estimator=LogisticRegression(), n_features_to_select=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	RFE(estimator=SVC(), n_features_to_select=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	RFE(estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	RFECV(estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.942352	0.928127	0.942352	0.93324	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.945265	0.928105	0.945265	0.932511	PCA(n_components=28)
HistGradientBoostingClassifier	0.945057	0.928057	0.945057	0.932597	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.944849	0.928024	0.944849	0.93268	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.945473	0.927942	0.945473	0.932187	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.934027	0.927832	0.934027	0.930713	MiniBatchSparsePCA(n_components=28)
StackingClassifier	0.945057	0.927823	0.945057	0.93237	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.944849	0.927787	0.944849	0.932455	SelectKBest(mutual_info_classif, k=28)
MLPClassifier	0.943809	0.927782	0.943809	0.932835	PCA(n_components=28)
GradientBoostingClassifier	0.945473	0.927714	0.945473	0.93195	MiniBatchSparsePCA(n_components=28)

MLPClassifier	0.946514	0.927628	0.946514	0.930356	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.946098	0.927625	0.946098	0.931133	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.945682	0.927579	0.945682	0.931607	MiniBatchSparsePCA(n_components=28)
SelfTrainingClassifier	0.945682	0.927579	0.945682	0.931607	MiniBatchSparsePCA(n_components=28)
GradientBoostingClassifier	0.944849	0.927548	0.944849	0.932228	PCA(n_components=28)
MLPClassifier	0.94589	0.927475	0.94589	0.931249	SelectKBest(f_classif, k=28)
MLPClassifier	0.945682	0.927351	0.945682	0.93136	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SGDClassifier	0.941727	0.927334	0.941727	0.932615	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.944017	0.927265	0.944017	0.932332	RFE(estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.925078	0.927253	0.925078	0.926147	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
HistGradientBoostingClassifier	0.944641	0.927031	0.944641	0.931857	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.941935	0.92699	0.941935	0.932365	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SGDClassifier	0.940895	0.926918	0.940895	0.932233	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GaussianNB	0.936316	0.926864	0.936316	0.93103	SelectKBest(f_classif, k=28)
GaussianNB	0.936316	0.926864	0.936316	0.93103	SelectKBest(mutual_info_classif, k=28)
GaussianNB	0.936316	0.926864	0.936316	0.93103	SelectKBest(f_regression, k=28)
GaussianNB	0.936316	0.926864	0.936316	0.93103	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GaussianNB	0.936316	0.926864	0.936316	0.93103	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GaussianNB	0.936316	0.926864	0.936316	0.93103	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GaussianNB	0.936316	0.926864	0.936316	0.93103	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.936316	0.926864	0.936316	0.93103	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GaussianNB	0.936316	0.926864	0.936316	0.93103	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)

GaussianNB	0.936316	0.926864	0.936316	0.931 03	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.936316	0.926864	0.936316	0.931 03	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GaussianNB	0.936316	0.926864	0.936316	0.931 03	RFECV(estimator=SVC(), min_features_to_select=28)
GaussianNB	0.936316	0.926864	0.936316	0.931 03	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	SelectKBest(f_classif, k=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	SelectKBest(mutual_info_classif, k=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	SelectKBest(f_regression, k=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	RFE(estimator=SVC(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	RFECV(estimator=SVC(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	PCA(n_components=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	KernelPCA(kernel="linear", n_components=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	SparsePCA(n_components=28)
LinearDiscriminantAnalysis	0.939 23	0.926766	0.939 23	0.931828	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.944433	0.926761	0.944433	0.931717	KernelPCA(kernel="linear", n_components=28)

SGDClassifier	0.946 93	0.926 753	0.946 93	0.927 09	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.946 722	0.926 651	0.946 722	0.928 204	KernelPCA(kernel="linear", n_components=28)
ExtraTreeClassifier	0.930 697	0.926 624	0.930 697	0.928 582	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.946 306	0.926 543	0.946 306	0.929 395	KernelPCA(kernel="linear", n_components=28)
LogisticRegression	0.946 306	0.926 543	0.946 306	0.929 395	SparsePCA(n_components=28)
LogisticRegression	0.946 306	0.926 543	0.946 306	0.929 395	MiniBatchSparsePCA(n_components=28)
LabelPropagation	0.946 306	0.926 543	0.946 306	0.929 395	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.943 809	0.926 514	0.943 809	0.931 748	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.937 981	0.926 49	0.937 981	0.931 325	SelectKBest(mutual_info_classif, k=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(f_classif, k=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(f_classif, k=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(mutual_info_classif, k=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(mutual_info_classif, k=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(f_regression, k=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	SelectKBest(f_regression, k=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SelfTrainingClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=LogisticRegression(), n_features_to_select=28)
KNeighborsClassifier	0.941 935	0.926 459	0.941 935	0.931 964	RFE(estimator=SVC(), n_features_to_select=28)

SelfTrainingClassifier	0.941935	0.926459	0.941935	0.931964	RFE(estimator=SVC(), n_features_to_select=28)
KNeighborsClassifier	0.941935	0.926459	0.941935	0.931964	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.941935	0.926459	0.941935	0.931964	RFE(estimator=MLPClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.941935	0.926459	0.941935	0.931964	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.941935	0.926459	0.941935	0.931964	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.941935	0.926459	0.941935	0.931964	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.941935	0.926459	0.941935	0.931964	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.941935	0.926459	0.941935	0.931964	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SelfTrainingClassifier	0.941935	0.926459	0.941935	0.931964	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
KNeighborsClassifier	0.941935	0.926459	0.941935	0.931964	RFECV(estimator=SVC(), min_features_to_select=28)
SelfTrainingClassifier	0.941935	0.926459	0.941935	0.931964	RFECV(estimator=SVC(), min_features_to_select=28)
KNeighborsClassifier	0.941935	0.926459	0.941935	0.931964	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.941935	0.926459	0.941935	0.931964	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.941935	0.926459	0.941935	0.931964	PCA(n_components=28)
SelfTrainingClassifier	0.941935	0.926459	0.941935	0.931964	PCA(n_components=28)
HistGradientBoostingClassifier	0.945265	0.926441	0.945265	0.930834	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SGDClassifier	0.94589	0.926332	0.94589	0.929949	SelectKBest(f_classif, k=28)
MLPClassifier	0.945473	0.926291	0.945473	0.930465	RFECV(estimator=SVC(), min_features_to_select=28)
HistGradientBoostingClassifier	0.944641	0.926273	0.944641	0.931152	PCA(n_components=28)
MLPClassifier	0.944433	0.926249	0.944433	0.93125	MiniBatchSparsePCA(n_components=28)
GaussianNB	0.928824	0.926149	0.928824	0.927456	PCA(n_components=28)
HistGradientBoostingClassifier	0.943392	0.926039	0.943392	0.931467	RFE(estimator=LogisticRegression(), n_features_to_select=28)

LabelSpreading	0.946514	0.926022	0.946514	0.928075	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.943809	0.925986	0.943809	0.931296	RFECV(estimator=SVC(), min_features_to_select=28)
MLPClassifier	0.944225	0.925975	0.944225	0.931111	SparsePCA(n_components=28)
KNeighborsClassifier	0.945057	0.925857	0.945057	0.930445	KernelPCA(kernel="linear", n_components=28)
SelfTrainingClassifier	0.945057	0.925857	0.945057	0.930445	KernelPCA(kernel="linear", n_components=28)
KNeighborsClassifier	0.945057	0.925857	0.945057	0.930445	SparsePCA(n_components=28)
SelfTrainingClassifier	0.945057	0.925857	0.945057	0.930445	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.919459	0.925824	0.919459	0.922512	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.9282	0.925638	0.9282	0.926891	KernelPCA(kernel="linear", n_components=28)
GaussianNB	0.9282	0.925638	0.9282	0.926891	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.936524	0.925612	0.936524	0.930328	RFECV(estimator=SVC(), min_features_to_select=28)
SGDClassifier	0.939438	0.925543	0.939438	0.931045	RFECV(estimator=SVC(), min_features_to_select=28)
SGDClassifier	0.93923	0.925393	0.93923	0.930903	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelPropagation	0.94589	0.925357	0.94589	0.928848	KernelPCA(kernel="linear", n_components=28)
LabelPropagation	0.94589	0.925357	0.94589	0.928848	SparsePCA(n_components=28)
SGDClassifier	0.941311	0.925356	0.941311	0.931129	PCA(n_components=28)
PassiveAggressiveClassifier	0.937357	0.925271	0.937357	0.930366	SelectKBest(f_classif, k=28)
MLPClassifier	0.9436	0.925189	0.9436	0.930694	RFE(estimator=LogisticRegression(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	SelectKBest(f_classif, k=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	SelectKBest(mutual_info_classif, k=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	SelectKBest(f_regression, k=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	RFE(estimator=LogisticRegression(), n_features_to_select=28)

QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	RFE(estimator=SVC(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	RFE(estimator=MLPClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	RFECV(estimator=SVC(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	PCA(n_components=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	KernelPCA(kernel="linear", n_components=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	SparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.926951	0.925177	0.926951	0.926052	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.945473	0.925007	0.945473	0.929138	MiniBatchSparsePCA(n_components=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	SelectKBest(f_classif, k=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	SelectKBest(mutual_info_classif, k=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	SelectKBest(f_regression, k=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	RFE(estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

LabelSpreading	0.942976	0.924739	0.942976	0.930509	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	RFECV(estimator=SVC(), min_features_to_select=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelSpreading	0.942976	0.924739	0.942976	0.930509	PCA(n_components=28)
GradientBoostingClassifier	0.944641	0.924655	0.944641	0.929662	SelectKBest(f_classif, k=28)
GradientBoostingClassifier	0.944641	0.924655	0.944641	0.929662	SelectKBest(mutual_info_classif, k=28)
GradientBoostingClassifier	0.944641	0.924655	0.944641	0.929662	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.944641	0.924655	0.944641	0.929662	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.944641	0.924655	0.944641	0.929662	RFE(estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.944641	0.924655	0.944641	0.929662	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.944641	0.924655	0.944641	0.929662	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.944641	0.924655	0.944641	0.929662	RFECV(estimator=SVC(), min_features_to_select=28)
GradientBoostingClassifier	0.944641	0.924655	0.944641	0.929662	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SGDClassifier	0.943392	0.924652	0.943392	0.93032	SelectKBest(mutual_info_classif, k=28)
SGDClassifier	0.943392	0.924652	0.943392	0.93032	SelectKBest(f_regression, k=28)
VotingClassifier	0.944225	0.9246	0.944225	0.929898	PCA(n_components=28)
LabelPropagation	0.94256	0.924571	0.94256	0.930459	SelectKBest(f_classif, k=28)
LabelPropagation	0.94256	0.924571	0.94256	0.930459	SelectKBest(mutual_info_classif, k=28)
LabelPropagation	0.94256	0.924571	0.94256	0.930459	SelectKBest(f_regression, k=28)
LabelPropagation	0.94256	0.924571	0.94256	0.930459	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelPropagation	0.94256	0.924571	0.94256	0.930459	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelPropagation	0.94256	0.924571	0.94256	0.930459	RFE(estimator=LogisticRegression(), n_features_to_select=28)

LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFE(estimator=SVC(), n_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFECV(estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelPropagation	0.942 56	0.924 571	0.942 56	0.930 459	PCA(n_components=28)
PassiveAggressiveClassifier	0.942 144	0.924 428	0.942 144	0.930 405	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.908 012	0.924 42	0.908 012	0.915 584	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.944 433	0.924 332	0.944 433	0.929 527	SelectKBest(f_regression, k=28)
GradientBoostingClassifier	0.944 433	0.924 332	0.944 433	0.929 527	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.944 433	0.924 332	0.944 433	0.929 527	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.944 433	0.924 332	0.944 433	0.929 527	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.943 392	0.924 068	0.943 392	0.929 842	SelectKBest(f_classif, k=28)
PassiveAggressiveClassifier	0.928 408	0.924 043	0.928 408	0.926 147	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	SelectKBest(f_regression, k=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	RFE(estimator=LogisticRegression(), n_features_to_select=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	RFE(estimator=MLPClassifier(), n_features_to_select=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
VotingClassifier	0.944 225	0.924 016	0.944 225	0.929 391	RFECV(estimator=SVC(), min_features_to_select=28)

VotingClassifier	0.944017	0.92401	0.944017	0.92951	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
VotingClassifier	0.944017	0.92401	0.944017	0.92951	RFE(estimator=SVC(), n_features_to_select=28)
VotingClassifier	0.944017	0.92401	0.944017	0.92951	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	SelectKBest(f_classif, k=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	SelectKBest(f_classif, k=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	SelectKBest(mutual_info_classif, k=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	SelectKBest(mutual_info_classif, k=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	SelectKBest(f_regression, k=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	SelectKBest(f_regression, k=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	RFE(estimator=LogisticRegression(), n_features_to_select=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	RFE(estimator=SVC(), n_features_to_select=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	RFE(estimator=SVC(), n_features_to_select=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	RFE(estimator=MLPClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)

LogisticRegression	0.945473	0.923877	0.945473	0.928014	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	RFECV(estimator=SVC(), min_features_to_select=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	RFECV(estimator=SVC(), min_features_to_select=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.945473	0.923877	0.945473	0.928014	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.945473	0.923877	0.945473	0.928014	PCA(n_components=28)
SGDClassifier	0.9436	0.923737	0.9436	0.929489	SparsePCA(n_components=28)
VotingClassifier	0.944017	0.923709	0.944017	0.929256	SelectKBest(f_classif, k=28)
VotingClassifier	0.944017	0.923709	0.944017	0.929256	SelectKBest(mutual_info_classif, k=28)
VotingClassifier	0.944017	0.923709	0.944017	0.929256	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.944017	0.923709	0.944017	0.929256	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.926951	0.923494	0.926951	0.925176	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.941311	0.923309	0.941311	0.929628	KernelPCA(kernel="linear", n_components=28)
SGDClassifier	0.946722	0.923098	0.946722	0.92122	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SGDClassifier	0.942976	0.922934	0.942976	0.929081	MiniBatchSparsePCA(n_components=28)
AdaBoostClassifier	0.94256	0.921763	0.94256	0.928314	KernelPCA(kernel="linear", n_components=28)
AdaBoostClassifier	0.94256	0.921763	0.94256	0.928314	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.936941	0.921745	0.936941	0.928002	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.920083	0.921539	0.920083	0.920805	KernelPCA(kernel="linear", n_components=28)

SGDClassifier	0.946098	0.921184	0.946098	0.924263	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.7923	0.920367	0.7923	0.844321	MiniBatchSparsePCA(n_components=28)
SVC	0.943809	0.919854	0.943809	0.926403	SelectKBest(f_classif, k=28)
SVC	0.943809	0.919854	0.943809	0.926403	SelectKBest(mutual_info_classif, k=28)
SVC	0.943809	0.919854	0.943809	0.926403	SelectKBest(f_regression, k=28)
SVC	0.943809	0.919854	0.943809	0.926403	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SVC	0.943809	0.919854	0.943809	0.926403	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SVC	0.943809	0.919854	0.943809	0.926403	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SVC	0.943809	0.919854	0.943809	0.926403	RFE(estimator=SVC(), n_features_to_select=28)
SVC	0.943809	0.919854	0.943809	0.926403	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SVC	0.943809	0.919854	0.943809	0.926403	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.943809	0.919854	0.943809	0.926403	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SVC	0.943809	0.919854	0.943809	0.926403	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SVC	0.943809	0.919854	0.943809	0.926403	RFECV(estimator=SVC(), min_features_to_select=28)
SVC	0.943809	0.919854	0.943809	0.926403	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.943809	0.919854	0.943809	0.926403	PCA(n_components=28)
SGDClassifier	0.94589	0.919628	0.94589	0.923795	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.867638	0.919611	0.867638	0.890431	PCA(n_components=28)
PassiveAggressiveClassifier	0.942768	0.91945	0.942768	0.926606	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SVC	0.944017	0.919374	0.944017	0.925942	KernelPCA(kernel="linear", n_components=28)
SVC	0.944017	0.919374	0.944017	0.925942	SparsePCA(n_components=28)
BernoulliNB	0.776483	0.917857	0.776483	0.833817	KernelPCA(kernel="linear", n_components=28)

BernoulliNB	0.776483	0.917857	0.776483	0.833817	SparsePCA(n_components=28)
SVC	0.9436	0.917105	0.9436	0.924782	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.945265	0.915206	0.945265	0.922737	RFECV(estimator=MLPClassifier(), min_features_to_select=28)

APPENDIX 7: VULNERABILITY FILE LEVEL RESULTS

Model	Accuracy	Precision	Recall	F1 Score	Feature Selection Model
RandomForestClassifier	0.952767	0.933185	0.952767	0.934382	MiniBatchSparsePCA(n_components=28)
MLPClassifier	0.952702	0.932658	0.952702	0.934027	MiniBatchSparsePCA(n_components=28)
KNeighborsClassifier	0.952178	0.930456	0.952178	0.933722	MiniBatchSparsePCA(n_components=28)
SelfTrainingClassifier	0.952178	0.930456	0.952178	0.933722	MiniBatchSparsePCA(n_components=28)
BaggingClassifier	0.951721	0.929007	0.951721	0.933 56	MiniBatchSparsePCA(n_components=28)
XGBClassifier	0.953 16	0.934976	0.953 16	0.933 42	MiniBatchSparsePCA(n_components=28)
StackingClassifier	0.953749	0.941 71	0.953749	0.933298	MiniBatchSparsePCA(n_components=28)
LabelPropagation	0.953094	0.934575	0.953094	0.932587	MiniBatchSparsePCA(n_components=28)
LabelSpreading	0.953291	0.937074	0.953291	0.932462	MiniBatchSparsePCA(n_components=28)
LGBMClassifier	0.953552	0.942369	0.953552	0.932369	MiniBatchSparsePCA(n_components=28)
HistGradientBoostingClassifier	0.953552	0.947049	0.953552	0.931886	MiniBatchSparsePCA(n_components=28)
VotingClassifier	0.953094	0.936793	0.953094	0.931153	MiniBatchSparsePCA(n_components=28)
SVC	0.952898	0.931679	0.952898	0.930172	MiniBatchSparsePCA(n_components=28)
BernoulliNB	0.952898	0.908015	0.952898	0.929915	MiniBatchSparsePCA(n_components=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	MiniBatchSparsePCA(n_components=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	MiniBatchSparsePCA(n_components=28)
LogisticRegression	0.952833	0.908012	0.952833	0.929882	MiniBatchSparsePCA(n_components=28)
LinearDiscriminantAnalysis	0.951917	0.914869	0.951917	0.929801	MiniBatchSparsePCA(n_components=28)
AdaBoostClassifier	0.952571	0.908	0.952571	0.929752	MiniBatchSparsePCA(n_components=28)
GradientBoostingClassifier	0.952571	0.908	0.952571	0.929752	MiniBatchSparsePCA(n_components=28)
SGDClassifier	0.951197	0.909677	0.951197	0.929189	MiniBatchSparsePCA(n_components=28)
PassiveAggressiveClassifier	0.931113	0.910178	0.931113	0.920 34	MiniBatchSparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.917441	0.916 56	0.917441	0.917	MiniBatchSparsePCA(n_components=28)
ExtraTreeClassifier	0.909 59	0.919416	0.909 59	0.914379	MiniBatchSparsePCA(n_components=28)
DecisionTreeClassifier	0.904 88	0.917969	0.904 88	0.911234	MiniBatchSparsePCA(n_components=28)

GaussianNB	0.900432	0.914744	0.900432	0.907412	MiniBatchSparsePCA(n_components=28)
XGBClassifier	0.954076	0.94006	0.954076	0.935788	PCA(n_components=28)
MLPClassifier	0.95316	0.935364	0.95316	0.935641	PCA(n_components=28)
StackingClassifier	0.953487	0.936785	0.953487	0.935226	PCA(n_components=28)
BaggingClassifier	0.952048	0.931448	0.952048	0.934869	PCA(n_components=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	PCA(n_components=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	PCA(n_components=28)
RandomForestClassifier	0.952506	0.932545	0.952506	0.934743	PCA(n_components=28)
LGBMClassifier	0.954337	0.946339	0.954337	0.93443	PCA(n_components=28)
HistGradientBoostingClassifier	0.953879	0.943118	0.953879	0.933487	PCA(n_components=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	PCA(n_components=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	PCA(n_components=28)
VotingClassifier	0.953356	0.940998	0.953356	0.931781	PCA(n_components=28)
GradientBoostingClassifier	0.953094	0.937461	0.953094	0.931029	PCA(n_components=28)
SVC	0.953094	0.94195	0.953094	0.930653	PCA(n_components=28)
AdaBoostClassifier	0.952898	0.931735	0.952898	0.9303	PCA(n_components=28)
BernoulliNB	0.952898	0.908015	0.952898	0.929915	PCA(n_components=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	PCA(n_components=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	PCA(n_components=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	PCA(n_components=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	PCA(n_components=28)
SGDClassifier	0.94544	0.910652	0.94544	0.926939	PCA(n_components=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	PCA(n_components=28)
ExtraTreeClassifier	0.911946	0.918971	0.911946	0.915394	PCA(n_components=28)
DecisionTreeClassifier	0.910114	0.918996	0.910114	0.914455	PCA(n_components=28)
GaussianNB	0.901086	0.914615	0.901086	0.907693	PCA(n_components=28)
PassiveAggressiveClassifier	0.851106	0.915312	0.851106	0.880618	PCA(n_components=28)

RandomForestClassifier	0.952963	0.935344	0.952963	0.936874	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BaggingClassifier	0.952178	0.933483	0.952178	0.936656	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
StackingClassifier	0.954337	0.941642	0.954337	0.93605	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.953945	0.943551	0.953945	0.933639	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
VotingClassifier	0.953749	0.945933	0.953749	0.932595	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
MLPClassifier	0.953094	0.934575	0.953094	0.932587	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.952963	0.933776	0.952963	0.930837	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
SGDClassifier	0.952636	0.91591	0.952636	0.929913	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)

LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.924572	0.914815	0.924572	0.919579	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.914366	0.92057	0.914366	0.917411	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.908282	0.920494	0.908282	0.914189	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.952571	0.934025	0.952571	0.936347	RFE(estimator=LogisticRegression(), n_features_to_select=28)
StackingClassifier	0.953618	0.937408	0.953618	0.935615	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BaggingClassifier	0.951328	0.930165	0.951328	0.934919	RFE(estimator=LogisticRegression(), n_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFE(estimator=LogisticRegression(), n_features_to_select=28)
HistGradientBoostingClassifier	0.953814	0.943842	0.953814	0.933103	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFE(estimator=LogisticRegression(), n_features_to_select=28)
VotingClassifier	0.953749	0.945933	0.953749	0.932595	RFE(estimator=LogisticRegression(), n_features_to_select=28)

LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFE(estimator=LogisticRegression(), n_features_to_select=28)
MLPClassifier	0.952506	0.929794	0.952506	0.932262	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GradientBoostingClassifier	0.952833	0.930388	0.952833	0.930769	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFE(estimator=LogisticRegression(), n_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFE(estimator=LogisticRegression(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFE(estimator=LogisticRegression(), n_features_to_select=28)
SGDClassifier	0.952178	0.911662	0.952178	0.929682	RFE(estimator=LogisticRegression(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFE(estimator=LogisticRegression(), n_features_to_select=28)
ExtraTreeClassifier	0.913319	0.921777	0.913319	0.917437	RFE(estimator=LogisticRegression(), n_features_to_select=28)
DecisionTreeClassifier	0.90789	0.920533	0.90789	0.913999	RFE(estimator=LogisticRegression(), n_features_to_select=28)
PassiveAggressiveClassifier	0.91188	0.914316	0.91188	0.913093	RFE(estimator=LogisticRegression(), n_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFE(estimator=LogisticRegression(), n_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFE(estimator=LogisticRegression(), n_features_to_select=28)
RandomForestClassifier	0.952636	0.934485	0.952636	0.936761	RFE(estimator=MLPClassifier(), n_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFE(estimator=MLPClassifier(), n_features_to_select=28)
StackingClassifier	0.954403	0.941777	0.954403	0.936297	RFE(estimator=MLPClassifier(), n_features_to_select=28)

BaggingClassifier	0.952178	0.932796	0.952178	0.935915	RFE(estimator=MLPClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFE(estimator=MLPClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.954599	0.949567	0.954599	0.934694	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFE(estimator=MLPClassifier(), n_features_to_select=28)
VotingClassifier	0.953683	0.946636	0.953683	0.93232	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.953029	0.935892	0.953029	0.930871	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFE(estimator=MLPClassifier(), n_features_to_select=28)
MLPClassifier	0.952767	0.928535	0.952767	0.930611	RFE(estimator=MLPClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFE(estimator=MLPClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFE(estimator=MLPClassifier(), n_features_to_select=28)
SGDClassifier	0.952506	0.907997	0.952506	0.929719	RFE(estimator=MLPClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFE(estimator=MLPClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.909721	0.919732	0.909721	0.914595	RFE(estimator=MLPClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.908413	0.9209	0.908413	0.914443	RFE(estimator=MLPClassifier(), n_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFE(estimator=MLPClassifier(), n_features_to_select=28)
PassiveAggressiveClassifier	0.802564	0.917364	0.802564	0.852567	RFE(estimator=MLPClassifier(), n_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFE(estimator=MLPClassifier(), n_features_to_select=28)
RandomForestClassifier	0.95316	0.935981	0.95316	0.937091	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)

StackingClassifier	0.954206	0.940351	0.954206	0.936381	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BaggingClassifier	0.951393	0.93113	0.951393	0.935616	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
HistGradientBoostingClassifier	0.95401	0.943467	0.95401	0.933904	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
MLPClassifier	0.95159	0.927114	0.95159	0.932638	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
VotingClassifier	0.953683	0.945408	0.953683	0.93244	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GradientBoostingClassifier	0.952963	0.933776	0.952963	0.930837	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
SGDClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)

PassiveAggressiveClassifier	0.945244	0.915889	0.945244	0.928587	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
ExtraTreeClassifier	0.9126	0.920477	0.9126	0.916449	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
DecisionTreeClassifier	0.906843	0.920381	0.906843	0.913374	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFE(estimator=RandomForestClassifier(), n_features_to_select=28)
RandomForestClassifier	0.952963	0.935617	0.952963	0.937335	RFE(estimator=SVC(), n_features_to_select=28)
BaggingClassifier	0.952636	0.934485	0.952636	0.936761	RFE(estimator=SVC(), n_features_to_select=28)
StackingClassifier	0.954272	0.940539	0.954272	0.936624	RFE(estimator=SVC(), n_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFE(estimator=SVC(), n_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFE(estimator=SVC(), n_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFE(estimator=SVC(), n_features_to_select=28)
MLPClassifier	0.953291	0.93576	0.953291	0.934478	RFE(estimator=SVC(), n_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFE(estimator=SVC(), n_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFE(estimator=SVC(), n_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFE(estimator=SVC(), n_features_to_select=28)
VotingClassifier	0.953749	0.947144	0.953749	0.932475	RFE(estimator=SVC(), n_features_to_select=28)
HistGradientBoostingClassifier	0.953291	0.939384	0.953291	0.931746	RFE(estimator=SVC(), n_features_to_select=28)
GradientBoostingClassifier	0.953029	0.935892	0.953029	0.930871	RFE(estimator=SVC(), n_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFE(estimator=SVC(), n_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=SVC(), n_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFE(estimator=SVC(), n_features_to_select=28)
SGDClassifier	0.952898	0.908015	0.952898	0.929915	RFE(estimator=SVC(), n_features_to_select=28)

LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFE(estimator=SVC(), n_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFE(estimator=SVC(), n_features_to_select=28)
ExtraTreeClassifier	0.914955	0.921764	0.914955	0.918285	RFE(estimator=SVC(), n_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFE(estimator=SVC(), n_features_to_select=28)
DecisionTreeClassifier	0.90802	0.920262	0.90802	0.913944	RFE(estimator=SVC(), n_features_to_select=28)
PassiveAggressiveClassifier	0.902787	0.918382	0.902787	0.910315	RFE(estimator=SVC(), n_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFE(estimator=SVC(), n_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFE(estimator=SVC(), n_features_to_select=28)
RandomForestClassifier	0.952702	0.934531	0.952702	0.936616	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BaggingClassifier	0.952178	0.933316	0.952178	0.936474	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
StackingClassifier	0.953749	0.937988	0.953749	0.936	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
MLPClassifier	0.95316	0.935178	0.95316	0.932739	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
VotingClassifier	0.953683	0.945408	0.953683	0.93244	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.953552	0.943205	0.953552	0.932249	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)

GradientBoostingClassifier	0.953029	0.935892	0.953029	0.930871	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
SGDClassifier	0.952833	0.908012	0.952833	0.929882	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.94616	0.910079	0.94616	0.927093	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.910506	0.920442	0.910506	0.915337	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.909329	0.91987	0.909329	0.914453	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=28)
RandomForestClassifier	0.952702	0.934597	0.952702	0.93671	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
StackingClassifier	0.954076	0.939864	0.954076	0.935995	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BaggingClassifier	0.951982	0.932482	0.951982	0.935981	RFECV(estimator=LogisticRegression(), min_features_to_select=28)

KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
HistGradientBoostingClassifier	0.954076	0.945558	0.954076	0.933713	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
VotingClassifier	0.953618	0.944839	0.953618	0.932284	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
MLPClassifier	0.952963	0.933233	0.952963	0.931692	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GradientBoostingClassifier	0.952898	0.931961	0.952898	0.930803	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
SGDClassifier	0.947992	0.911895	0.947992	0.92826	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
ExtraTreeClassifier	0.909721	0.920224	0.909721	0.914823	RFECV(estimator=LogisticRegression(), min_features_to_select=28)

DecisionTreeClassifier	0.908675	0.921036	0.908675	0.914644	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
PassiveAggressiveClassifier	0.88931	0.914008	0.88931	0.901208	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFECV(estimator=LogisticRegression(), min_features_to_select=28)
RandomForestClassifier	0.952833	0.934783	0.952833	0.93651	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
StackingClassifier	0.953879	0.938808	0.953879	0.935773	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
BaggingClassifier	0.95087	0.929688	0.95087	0.935016	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.953945	0.947293	0.953945	0.933058	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
VotingClassifier	0.953749	0.947144	0.953749	0.932475	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
MLPClassifier	0.953225	0.936224	0.953225	0.932426	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.952963	0.933776	0.952963	0.930837	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFECV(estimator=MLPClassifier(), min_features_to_select=28)

AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
SGDClassifier	0.952375	0.915955	0.952375	0.929908	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
PassiveAggressiveClassifier	0.93844	0.916595	0.93844	0.926575	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.914759	0.922609	0.914759	0.918582	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.908086	0.920078	0.908086	0.913894	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFECV(estimator=MLPClassifier(), min_features_to_select=28)
RandomForestClassifier	0.952506	0.934139	0.952506	0.936679	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BaggingClassifier	0.952244	0.932882	0.952244	0.935861	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
StackingClassifier	0.953618	0.937432	0.953618	0.935512	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
VotingClassifier	0.953814	0.947612	0.953814	0.932631	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
HistGradientBoostingClassifier	0.953683	0.950056	0.953683	0.932078	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
MLPClassifier	0.952898	0.932074	0.952898	0.93105	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GradientBoostingClassifier	0.952963	0.933776	0.952963	0.930837	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
SGDClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
ExtraTreeClassifier	0.912404	0.920048	0.912404	0.916144	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
DecisionTreeClassifier	0.90946	0.920962	0.90946	0.915026	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)

PassiveAggressiveClassifier	0.845022	0.918362	0.845022	0.878035	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFECV(estimator=RandomForestClassifier(), min_features_to_select=28)
RandomForestClassifier	0.953356	0.936699	0.953356	0.937493	RFECV(estimator=SVC(), min_features_to_select=28)
BaggingClassifier	0.952636	0.934417	0.952636	0.936669	RFECV(estimator=SVC(), min_features_to_select=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	RFECV(estimator=SVC(), min_features_to_select=28)
StackingClassifier	0.954076	0.940171	0.954076	0.935684	RFECV(estimator=SVC(), min_features_to_select=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=SVC(), min_features_to_select=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	RFECV(estimator=SVC(), min_features_to_select=28)
HistGradientBoostingClassifier	0.954206	0.944595	0.954206	0.934355	RFECV(estimator=SVC(), min_features_to_select=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	RFECV(estimator=SVC(), min_features_to_select=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	RFECV(estimator=SVC(), min_features_to_select=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	RFECV(estimator=SVC(), min_features_to_select=28)
MLPClassifier	0.953225	0.936077	0.953225	0.932543	RFECV(estimator=SVC(), min_features_to_select=28)
VotingClassifier	0.953618	0.944839	0.953618	0.932284	RFECV(estimator=SVC(), min_features_to_select=28)
GradientBoostingClassifier	0.953029	0.935892	0.953029	0.930871	RFECV(estimator=SVC(), min_features_to_select=28)
SVC	0.953094	0.94195	0.953094	0.930653	RFECV(estimator=SVC(), min_features_to_select=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=SVC(), min_features_to_select=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	RFECV(estimator=SVC(), min_features_to_select=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	RFECV(estimator=SVC(), min_features_to_select=28)
SGDClassifier	0.952375	0.912758	0.952375	0.929781	RFECV(estimator=SVC(), min_features_to_select=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	RFECV(estimator=SVC(), min_features_to_select=28)
PassiveAggressiveClassifier	0.948515	0.910556	0.948515	0.928187	RFECV(estimator=SVC(), min_features_to_select=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	RFECV(estimator=SVC(), min_features_to_select=28)
DecisionTreeClassifier	0.908936	0.921075	0.908936	0.914801	RFECV(estimator=SVC(), min_features_to_select=28)
ExtraTreeClassifier	0.909329	0.91898	0.909329	0.914038	RFECV(estimator=SVC(), min_features_to_select=28)

GaussianNB	0.890684	0.916518	0.890684	0.90302	RFECV(estimator=SVC(), min_features_to_select=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	RFECV(estimator=SVC(), min_features_to_select=28)
RandomForestClassifier	0.952898	0.935153	0.952898	0.936833	SelectKBest(f_classif, k=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	SelectKBest(f_classif, k=28)
BaggingClassifier	0.951786	0.931911	0.951786	0.935766	SelectKBest(f_classif, k=28)
StackingClassifier	0.953094	0.934867	0.953094	0.934992	SelectKBest(f_classif, k=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	SelectKBest(f_classif, k=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	SelectKBest(f_classif, k=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	SelectKBest(f_classif, k=28)
MLPClassifier	0.953356	0.936966	0.953356	0.933078	SelectKBest(f_classif, k=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	SelectKBest(f_classif, k=28)
HistGradientBoostingClassifier	0.953749	0.944925	0.953749	0.932714	SelectKBest(f_classif, k=28)
VotingClassifier	0.953749	0.945933	0.953749	0.932595	SelectKBest(f_classif, k=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	SelectKBest(f_classif, k=28)
GradientBoostingClassifier	0.953029	0.935892	0.953029	0.930871	SelectKBest(f_classif, k=28)
SVC	0.953094	0.94195	0.953094	0.930653	SelectKBest(f_classif, k=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	SelectKBest(f_classif, k=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	SelectKBest(f_classif, k=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	SelectKBest(f_classif, k=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	SelectKBest(f_classif, k=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	SelectKBest(f_classif, k=28)
PassiveAggressiveClassifier	0.949169	0.916399	0.949169	0.929663	SelectKBest(f_classif, k=28)
SGDClassifier	0.943609	0.913114	0.943609	0.927007	SelectKBest(f_classif, k=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	SelectKBest(f_classif, k=28)
ExtraTreeClassifier	0.912273	0.921505	0.912273	0.916761	SelectKBest(f_classif, k=28)
DecisionTreeClassifier	0.906647	0.92016	0.906647	0.913169	SelectKBest(f_classif, k=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	SelectKBest(f_classif, k=28)

BernoulliNB	0.742117	0.920026	0.742117	0.814678	SelectKBest(f_classif, k=28)
RandomForestClassifier	0.95316	0.936026	0.95316	0.937184	SelectKBest(f_regression, k=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	SelectKBest(f_regression, k=28)
StackingClassifier	0.953945	0.93899	0.953945	0.93612	SelectKBest(f_regression, k=28)
BaggingClassifier	0.952178	0.932706	0.952178	0.93582	SelectKBest(f_regression, k=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	SelectKBest(f_regression, k=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	SelectKBest(f_regression, k=28)
HistGradientBoostingClassifier	0.954337	0.947664	0.954337	0.934204	SelectKBest(f_regression, k=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	SelectKBest(f_regression, k=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	SelectKBest(f_regression, k=28)
MLPClassifier	0.953487	0.939561	0.953487	0.932688	SelectKBest(f_regression, k=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	SelectKBest(f_regression, k=28)
VotingClassifier	0.953683	0.945408	0.953683	0.93244	SelectKBest(f_regression, k=28)
GradientBoostingClassifier	0.952963	0.933776	0.952963	0.930837	SelectKBest(f_regression, k=28)
SVC	0.953094	0.94195	0.953094	0.930653	SelectKBest(f_regression, k=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	SelectKBest(f_regression, k=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	SelectKBest(f_regression, k=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	SelectKBest(f_regression, k=28)
SGDClassifier	0.952898	0.908015	0.952898	0.929915	SelectKBest(f_regression, k=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	SelectKBest(f_regression, k=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	SelectKBest(f_regression, k=28)
PassiveAggressiveClassifier	0.935824	0.91246	0.935824	0.923538	SelectKBest(f_regression, k=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	SelectKBest(f_regression, k=28)
ExtraTreeClassifier	0.908805	0.919891	0.908805	0.914187	SelectKBest(f_regression, k=28)
DecisionTreeClassifier	0.908544	0.920047	0.908544	0.914122	SelectKBest(f_regression, k=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	SelectKBest(f_regression, k=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	SelectKBest(f_regression, k=28)

BaggingClassifier	0.952178	0.9334	0.952178	0.936565	SelectKBest(mutual_info_classif, k=28)
RandomForestClassifier	0.952767	0.934652	0.952767	0.936564	SelectKBest(mutual_info_classif, k=28)
XGBClassifier	0.954795	0.944824	0.954795	0.936433	SelectKBest(mutual_info_classif, k=28)
StackingClassifier	0.953356	0.936111	0.953356	0.935044	SelectKBest(mutual_info_classif, k=28)
KNeighborsClassifier	0.953225	0.935437	0.953225	0.934862	SelectKBest(mutual_info_classif, k=28)
SelfTrainingClassifier	0.953225	0.935437	0.953225	0.934862	SelectKBest(mutual_info_classif, k=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	SelectKBest(mutual_info_classif, k=28)
LabelPropagation	0.952833	0.932442	0.952833	0.932785	SelectKBest(mutual_info_classif, k=28)
HistGradientBoostingClassifier	0.953421	0.938609	0.953421	0.932652	SelectKBest(mutual_info_classif, k=28)
VotingClassifier	0.953749	0.945933	0.953749	0.932595	SelectKBest(mutual_info_classif, k=28)
LabelSpreading	0.952898	0.932812	0.952898	0.932594	SelectKBest(mutual_info_classif, k=28)
MLPClassifier	0.953291	0.937074	0.953291	0.932462	SelectKBest(mutual_info_classif, k=28)
GradientBoostingClassifier	0.953029	0.935892	0.953029	0.930871	SelectKBest(mutual_info_classif, k=28)
SVC	0.953094	0.94195	0.953094	0.930653	SelectKBest(mutual_info_classif, k=28)
AdaBoostClassifier	0.952898	0.908015	0.952898	0.929915	SelectKBest(mutual_info_classif, k=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	SelectKBest(mutual_info_classif, k=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	SelectKBest(mutual_info_classif, k=28)
LogisticRegression	0.952767	0.908009	0.952767	0.92985	SelectKBest(mutual_info_classif, k=28)
SGDClassifier	0.952702	0.908006	0.952702	0.929817	SelectKBest(mutual_info_classif, k=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	SelectKBest(mutual_info_classif, k=28)
PassiveAggressiveClassifier	0.947403	0.911051	0.947403	0.927844	SelectKBest(mutual_info_classif, k=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	SelectKBest(mutual_info_classif, k=28)
ExtraTreeClassifier	0.91188	0.922118	0.91188	0.916838	SelectKBest(mutual_info_classif, k=28)
DecisionTreeClassifier	0.907105	0.920226	0.907105	0.913442	SelectKBest(mutual_info_classif, k=28)
GaussianNB	0.890684	0.916518	0.890684	0.90302	SelectKBest(mutual_info_classif, k=28)
BernoulliNB	0.742117	0.920026	0.742117	0.814678	SelectKBest(mutual_info_classif, k=28)
MLPClassifier	0.95087	0.931093	0.95087	0.936101	SparsePCA(n_components=28)

XGBClassifier	0.954076	0.939864	0.954076	0.935995	SparsePCA(n_components=28)
RandomForestClassifier	0.952571	0.933113	0.952571	0.935185	SparsePCA(n_components=28)
StackingClassifier	0.953487	0.936911	0.953487	0.9347	SparsePCA(n_components=28)
BaggingClassifier	0.951786	0.930582	0.951786	0.934613	SparsePCA(n_components=28)
KNeighborsClassifier	0.952702	0.933054	0.952702	0.934551	SparsePCA(n_components=28)
SelfTrainingClassifier	0.952702	0.933054	0.952702	0.934551	SparsePCA(n_components=28)
LGBMClassifier	0.954272	0.946636	0.954272	0.934167	SparsePCA(n_components=28)
LabelPropagation	0.953291	0.936211	0.953291	0.933156	SparsePCA(n_components=28)
LabelSpreading	0.953356	0.937485	0.953356	0.932732	SparsePCA(n_components=28)
HistGradientBoostingClassifier	0.953749	0.948622	0.953749	0.932355	SparsePCA(n_components=28)
VotingClassifier	0.953421	0.940344	0.953421	0.932178	SparsePCA(n_components=28)
SVC	0.953356	0.943198	0.953356	0.931537	SparsePCA(n_components=28)
GradientBoostingClassifier	0.95316	0.93803	0.95316	0.931311	SparsePCA(n_components=28)
AdaBoostClassifier	0.952833	0.923769	0.952833	0.930011	SparsePCA(n_components=28)
BernoulliNB	0.952898	0.908015	0.952898	0.929915	SparsePCA(n_components=28)
RidgeClassifier	0.952898	0.908015	0.952898	0.929915	SparsePCA(n_components=28)
RidgeClassifierCV	0.952898	0.908015	0.952898	0.929915	SparsePCA(n_components=28)
LogisticRegression	0.952833	0.908012	0.952833	0.929882	SparsePCA(n_components=28)
LinearDiscriminantAnalysis	0.951851	0.91456	0.951851	0.929768	SparsePCA(n_components=28)
SGDClassifier	0.952309	0.912327	0.952309	0.929748	SparsePCA(n_components=28)
QuadraticDiscriminantAnalysis	0.917833	0.917068	0.917833	0.91745	SparsePCA(n_components=28)
ExtraTreeClassifier	0.91031	0.918925	0.91031	0.914523	SparsePCA(n_components=28)
DecisionTreeClassifier	0.908871	0.918415	0.908871	0.913533	SparsePCA(n_components=28)
PassiveAggressiveClassifier	0.906058	0.911219	0.906058	0.908621	SparsePCA(n_components=28)
GaussianNB	0.901086	0.914615	0.901086	0.907693	SparsePCA(n_components=28)

APPENDIX 8: AUTOML EXPERIMENT RESULTS

Model	ROC AUC	Precision	Recall	F1 Score	Feature Selection Model
PassiveAggressiveClassifier	0.592519	0.80084	0.201055	0.321417	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
PassiveAggressiveClassifier	0.638139	0.684023	0.324262	0.43996	SelectKBest(mutual_info_classif, k=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	SelectKBest(f_classif, k=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	SelectKBest(mutual_info_classif, k=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	SelectKBest(f_regression, k=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFE(estimator=LogisticRegression(), n_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFE(estimator=SVC(), n_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFE(estimator=MLPClassifier(), n_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFECV(estimator=SVC(), min_features_to_select=4)
GaussianNB	0.610679	0.671706	0.262447	0.377427	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
SVC	0.669785	0.670056	0.403165	0.503425	SparsePCA(n_components=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	SelectKBest(f_classif, k=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	SelectKBest(mutual_info_classif, k=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	SelectKBest(f_regression, k=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFE(estimator=LogisticRegression(), n_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFE(estimator=SVC(), n_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFE(estimator=MLPClassifier(), n_features_to_select=4)

QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFECV(estimator=SVC(), min_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	PCA(n_components=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	SparsePCA(n_components=4)
QuadraticDiscriminantAnalysis	0.601693	0.668611	0.241772	0.355129	MiniBatchSparsePCA(n_components=4)
PassiveAggressiveClassifier	0.632999	0.665339	0.317089	0.42949	SelectKBest(f_classif, k=4)
SVC	0.67237	0.665301	0.41097	0.508086	SelectKBest(f_classif, k=4)
SVC	0.67237	0.665301	0.41097	0.508086	SelectKBest(mutual_info_classif, k=4)
SVC	0.67237	0.665301	0.41097	0.508086	SelectKBest(f_regression, k=4)
SVC	0.67237	0.665301	0.41097	0.508086	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
SVC	0.67237	0.665301	0.41097	0.508086	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
SVC	0.67237	0.665301	0.41097	0.508086	RFE(estimator=LogisticRegression(), n_features_to_select=4)
SVC	0.67237	0.665301	0.41097	0.508086	RFE(estimator=SVC(), n_features_to_select=4)
SVC	0.67237	0.665301	0.41097	0.508086	RFE(estimator=MLPClassifier(), n_features_to_select=4)
SVC	0.67237	0.665301	0.41097	0.508086	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
SVC	0.67237	0.665301	0.41097	0.508086	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
SVC	0.67237	0.665301	0.41097	0.508086	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
SVC	0.67237	0.665301	0.41097	0.508086	RFECV(estimator=SVC(), min_features_to_select=4)
SVC	0.67237	0.665301	0.41097	0.508086	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
SVC	0.67237	0.665301	0.41097	0.508086	PCA(n_components=4)
SVC	0.671417	0.664837	0.408861	0.506336	MiniBatchSparsePCA(n_components=4)
VotingClassifier	0.677701	0.658624	0.42616	0.517484	RFE(estimator=LogisticRegression(), n_features_to_select=4)
VotingClassifier	0.677351	0.658075	0.425527	0.516848	RFE(estimator=SVC(), n_features_to_select=4)
VotingClassifier	0.6776	0.65798	0.42616	0.517286	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)

VotingClassifier	0.677815	0.657672	0.426793	0.517656	SelectKBest(f_regression, k=4)
VotingClassifier	0.675938	0.657471	0.422363	0.514322	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
VotingClassifier	0.675942	0.656937	0.422574	0.514315	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
VotingClassifier	0.676191	0.656843	0.423207	0.514755	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
VotingClassifier	0.67768	0.656818	0.426793	0.517391	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
VotingClassifier	0.676937	0.656566	0.425105	0.516071	RFECV(estimator=SVC(), min_features_to_select=4)
VotingClassifier	0.677718	0.656503	0.427004	0.517449	RFE(estimator=MLPClassifier(), n_features_to_select=4)
VotingClassifier	0.677363	0.656484	0.42616	0.516822	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
VotingClassifier	0.677507	0.65628	0.426582	0.517069	SelectKBest(mutual_info_classif, k=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	SelectKBest(f_classif, k=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	SelectKBest(mutual_info_classif, k=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	SelectKBest(f_regression, k=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	RFE(estimator=LogisticRegression(), n_features_to_select=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	RFE(estimator=SVC(), n_features_to_select=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	RFE(estimator=MLPClassifier(), n_features_to_select=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	RFECV(estimator=SVC(), min_features_to_select=4)
BernoulliNB	0.639789	0.655825	0.336076	0.444413	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
VotingClassifier	0.677334	0.655743	0.426371	0.516748	SelectKBest(f_classif, k=4)
VotingClassifier	0.680688	0.636226	0.442405	0.521901	MiniBatchSparsePCA(n_components=4)
VotingClassifier	0.677172	0.62845	0.437131	0.515615	SparsePCA(n_components=4)
VotingClassifier	0.677075	0.627421	0.437342	0.515415	PCA(n_components=4)
StackingClassifier	0.645378	0.626794	0.359283	0.456752	RFE(estimator=LogisticRegression(), n_features_to_select=4)

StackingClassifier	0.642188	0.625986	0.351688	0.450358	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
StackingClassifier	0.649182	0.625984	0.368987	0.464295	SelectKBest(mutual_info_classif, k=4)
StackingClassifier	0.642863	0.625934	0.353376	0.451726	SparsePCA(n_components=4)
StackingClassifier	0.642724	0.625561	0.353165	0.451456	RFE(estimator=MLPClassifier(), n_features_to_select=4)
StackingClassifier	0.64855	0.625179	0.367722	0.463071	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
SGDClassifier	0.603531	0.625	0.256329	0.363555	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
RidgeClassifierCV	0.64809	0.624955	0.366667	0.462173	SparsePCA(n_components=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	SelectKBest(f_classif, k=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	SelectKBest(mutual_info_classif, k=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	SelectKBest(f_regression, k=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	RFE(estimator=LogisticRegression(), n_features_to_select=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	RFE(estimator=SVC(), n_features_to_select=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	RFE(estimator=MLPClassifier(), n_features_to_select=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	RFECV(estimator=SVC(), min_features_to_select=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
RidgeClassifierCV	0.649191	0.624911	0.369409	0.464333	PCA(n_components=4)
StackingClassifier	0.648161	0.624865	0.366878	0.462316	SelectKBest(f_classif, k=4)
StackingClassifier	0.646891	0.62378	0.364135	0.459837	SelectKBest(f_regression, k=4)
StackingClassifier	0.642631	0.623744	0.353586	0.451326	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
RidgeClassifierCV	0.648136	0.623567	0.3673	0.462294	MiniBatchSparsePCA(n_components=4)
StackingClassifier	0.639189	0.623476	0.345148	0.444324	PCA(n_components=4)
StackingClassifier	0.644883	0.62172	0.359916	0.455906	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)

StackingClassifier	0.645672	0.620752	0.362236	0.457501	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
StackingClassifier	0.652961	0.619636	0.380802	0.47171	RFE(estimator=SVC(), n_features_to_select=4)
StackingClassifier	0.647566	0.619268	0.367511	0.461274	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
StackingClassifier	0.646895	0.618759	0.366034	0.459968	RFECV(estimator=SVC(), min_features_to_select=4)
RidgeClassifier	0.652298	0.618132	0.379747	0.470465	SparsePCA(n_components=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	SelectKBest(f_classif, k=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	SelectKBest(mutual_info_classif, k=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	SelectKBest(f_regression, k=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	RFE(estimator=LogisticRegression(), n_features_to_select=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	RFE(estimator=SVC(), n_features_to_select=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	RFE(estimator=MLPClassifier(), n_features_to_select=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	RFECV(estimator=SVC(), min_features_to_select=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
RidgeClassifier	0.65237	0.618051	0.379958	0.470604	PCA(n_components=4)
RidgeClassifier	0.652336	0.617839	0.379958	0.470542	MiniBatchSparsePCA(n_components=4)
MLPClassifier	0.699653	0.617194	0.498312	0.551418	MiniBatchSparsePCA(n_components=4)
MLPClassifier	0.694506	0.616827	0.485654	0.543437	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
StackingClassifier	0.651568	0.616203	0.378692	0.469097	MiniBatchSparsePCA(n_components=4)
MLPClassifier	0.699243	0.615645	0.498101	0.550671	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
PassiveAggressiveClassifier	0.624402	0.615096	0.311181	0.413281	RFE(estimator=MLPClassifier(), n_features_to_select=4)
MLPClassifier	0.698754	0.614923	0.497257	0.549866	RFE(estimator=LogisticRegression(), n_features_to_select=4)
PassiveAggressiveClassifier	0.671368	0.614571	0.428903	0.505219	RFE(estimator=SVC(), n_features_to_select=4)

MLPClassifier	0.697382	0.613613	0.494515	0.547664	RFE(estimator=MLPClassifier(), n_features_to_select=4)
MLPClassifier	0.696353	0.613523	0.491983	0.546072	RFE(estimator=SVC(), n_features_to_select=4)
MLPClassifier	0.697956	0.613139	0.496203	0.548507	RFECV(estimator=SVC(), min_features_to_select=4)
MLPClassifier	0.696471	0.61248	0.492827	0.546177	PCA(n_components=4)
MLPClassifier	0.704278	0.610123	0.513713	0.557783	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
MLPClassifier	0.693619	0.610098	0.48692	0.541593	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
MLPClassifier	0.697689	0.608348	0.498101	0.547732	SparsePCA(n_components=4)
MLPClassifier	0.697157	0.608213	0.496835	0.546911	SelectKBest(f_regression, k=4)
MLPClassifier	0.695774	0.608006	0.49346	0.544777	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
MLPClassifier	0.698014	0.607914	0.499156	0.548193	SelectKBest(f_classif, k=4)
MLPClassifier	0.692758	0.607595	0.486076	0.540084	SelectKBest(mutual_info_classif, k=4)
MLPClassifier	0.701546	0.600544	0.512236	0.552886	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
PassiveAggressiveClassifier	0.616886	0.596876	0.298312	0.397806	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	SelectKBest(f_classif, k=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	SelectKBest(mutual_info_classif, k=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	SelectKBest(f_regression, k=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	RFE(estimator=SVC(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	RFE(estimator=MLPClassifier(), n_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	RFECV(estimator=SVC(), min_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	PCA(n_components=4)

LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	SparsePCA(n_components=4)
LinearDiscriminantAnalysis	0.668638	0.592261	0.4327	0.500061	MiniBatchSparsePCA(n_components=4)
LogisticRegression	0.673658	0.579061	0.452743	0.50817	SparsePCA(n_components=4)
LogisticRegression	0.673696	0.578862	0.452954	0.508226	MiniBatchSparsePCA(n_components=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	SelectKBest(f_classif, k=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	SelectKBest(mutual_info_classif, k=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	SelectKBest(f_regression, k=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	RFE(estimator=SVC(), n_features_to_select=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	RFE(estimator=MLPClassifier(), n_features_to_select=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	RFECV(estimator=SVC(), min_features_to_select=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
LogisticRegression	0.673885	0.577873	0.454008	0.508507	PCA(n_components=4)
PassiveAggressiveClassifier	0.575128	0.576828	0.196414	0.293044	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
GaussianNB	0.651 09	0.575284	0.395781	0.468941	PCA(n_components=4)
GaussianNB	0.651 09	0.575284	0.395781	0.468941	SparsePCA(n_components=4)
LabelPropagation	0.688 89	0.568477	0.499156	0.531566	SparsePCA(n_components=4)
LabelSpreading	0.688 89	0.568477	0.499156	0.531566	SparsePCA(n_components=4)
LabelSpreading	0.689641	0.568285	0.501266	0.532676	MiniBatchSparsePCA(n_components=4)
LabelPropagation	0.689607	0.568149	0.501266	0.532616	MiniBatchSparsePCA(n_components=4)
LabelPropagation	0.696232	0.563268	0.522152	0.541931	SelectKBest(mutual_info_classif, k=4)
LabelSpreading	0.696232	0.563268	0.522152	0.541931	SelectKBest(mutual_info_classif, k=4)

LabelPropagation	0.69527	0.563229	0.51962	0.540546	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
LabelSpreading	0.69527	0.563229	0.51962	0.540546	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
LabelPropagation	0.69527	0.563229	0.51962	0.540546	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LabelSpreading	0.69527	0.563229	0.51962	0.540546	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LabelPropagation	0.69527	0.563229	0.51962	0.540546	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LabelSpreading	0.69527	0.563229	0.51962	0.540546	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LabelPropagation	0.695342	0.5632	0.519831	0.540647	SelectKBest(f_regression, k=4)
LabelSpreading	0.695342	0.5632	0.519831	0.540647	SelectKBest(f_regression, k=4)
LabelPropagation	0.695455	0.562728	0.520464	0.540772	PCA(n_components=4)
LabelSpreading	0.695455	0.562728	0.520464	0.540772	PCA(n_components=4)
LabelSpreading	0.696505	0.56161	0.524051	0.542181	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
LabelSpreading	0.696505	0.56161	0.524051	0.542181	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LabelSpreading	0.696505	0.56161	0.524051	0.542181	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
LabelSpreading	0.696505	0.56161	0.524051	0.542181	RFECV(estimator=SVC(), min_features_to_select=4)
LabelSpreading	0.696505	0.56161	0.524051	0.542181	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
LabelPropagation	0.696471	0.561483	0.524051	0.542121	RFE(estimator=SVC(), n_features_to_select=4)
LabelSpreading	0.696471	0.561483	0.524051	0.542121	RFE(estimator=SVC(), n_features_to_select=4)
LabelPropagation	0.696471	0.561483	0.524051	0.542121	RFE(estimator=MLPClassifier(), n_features_to_select=4)
LabelSpreading	0.696471	0.561483	0.524051	0.542121	RFE(estimator=MLPClassifier(), n_features_to_select=4)
LabelPropagation	0.696471	0.561483	0.524051	0.542121	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
LabelPropagation	0.696471	0.561483	0.524051	0.542121	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LabelPropagation	0.696471	0.561483	0.524051	0.542121	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
LabelPropagation	0.696471	0.561483	0.524051	0.542121	RFECV(estimator=SVC(), min_features_to_select=4)
LabelPropagation	0.696471	0.561483	0.524051	0.542121	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
LabelPropagation	0.697437	0.56125	0.526793	0.543476	SelectKBest(f_classif, k=4)
LabelSpreading	0.697437	0.56125	0.526793	0.543476	SelectKBest(f_classif, k=4)
GradientBoostingClassifier	0.686464	0.553389	0.502954	0.526967	PCA(n_components=4)

GradientBoostingClassifier	0.685835	0.552485	0.501899	0.525978	SparsePCA(n_components=4)
SGDClassifier	0.682923	0.549125	0.496414	0.52144	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
GaussianNB	0.641652	0.548406	0.38481	0.452269	MiniBatchSparsePCA(n_components=4)
KNeighborsClassifier	0.687832	0.547738	0.510759	0.528603	MiniBatchSparsePCA(n_components=4)
SelfTrainingClassifier	0.687832	0.547738	0.510759	0.528603	MiniBatchSparsePCA(n_components=4)
SGDClassifier	0.641951	0.547221	0.386287	0.452882	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
GradientBoostingClassifier	0.68665	0.543434	0.510759	0.526591	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
GradientBoostingClassifier	0.686616	0.543312	0.510759	0.526533	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
GradientBoostingClassifier	0.686582	0.54319	0.510759	0.526476	SelectKBest(f_classif, k=4)
GradientBoostingClassifier	0.686654	0.543171	0.51097	0.526579	SelectKBest(mutual_info_classif, k=4)
GradientBoostingClassifier	0.68662	0.543049	0.51097	0.526522	RFE(estimator=LogisticRegression(), n_features_to_select=4)
GradientBoostingClassifier	0.686514	0.542947	0.510759	0.526362	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
GradientBoostingClassifier	0.686514	0.542947	0.510759	0.526362	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
GradientBoostingClassifier	0.686586	0.542928	0.51097	0.526465	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
GradientBoostingClassifier	0.686409	0.542844	0.510549	0.526201	RFE(estimator=MLPClassifier(), n_features_to_select=4)
GradientBoostingClassifier	0.686375	0.542723	0.510549	0.526144	RFE(estimator=SVC(), n_features_to_select=4)
GradientBoostingClassifier	0.686375	0.542723	0.510549	0.526144	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
GradientBoostingClassifier	0.686447	0.542703	0.510759	0.526247	SelectKBest(f_regression, k=4)
GradientBoostingClassifier	0.686413	0.542582	0.510759	0.52619	RFECV(estimator=SVC(), min_features_to_select=4)
AdaBoostClassifier	0.679289	0.54157	0.491983	0.515587	PCA(n_components=4)
AdaBoostClassifier	0.679289	0.54157	0.491983	0.515587	SparsePCA(n_components=4)
KNeighborsClassifier	0.693587	0.540506	0.532068	0.536253	RFE(estimator=SVC(), n_features_to_select=4)
SelfTrainingClassifier	0.693587	0.540506	0.532068	0.536253	RFE(estimator=SVC(), n_features_to_select=4)
KNeighborsClassifier	0.693587	0.540506	0.532068	0.536253	RFE(estimator=MLPClassifier(), n_features_to_select=4)
SelfTrainingClassifier	0.693587	0.540506	0.532068	0.536253	RFE(estimator=MLPClassifier(), n_features_to_select=4)
KNeighborsClassifier	0.693587	0.540506	0.532068	0.536253	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
SelfTrainingClassifier	0.693587	0.540506	0.532068	0.536253	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)

KNeighborsClassifier	0.693553	0.54039	0.532068	0.536196	SelectKBest(f_classif, k=4)
SelfTrainingClassifier	0.693553	0.54039	0.532068	0.536196	SelectKBest(f_classif, k=4)
KNeighborsClassifier	0.693553	0.54039	0.532068	0.536196	SelectKBest(f_regression, k=4)
SelfTrainingClassifier	0.693553	0.54039	0.532068	0.536196	SelectKBest(f_regression, k=4)
KNeighborsClassifier	0.693519	0.540274	0.532068	0.536139	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
SelfTrainingClassifier	0.693519	0.540274	0.532068	0.536139	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
KNeighborsClassifier	0.693519	0.540274	0.532068	0.536139	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
SelfTrainingClassifier	0.693519	0.540274	0.532068	0.536139	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
KNeighborsClassifier	0.693519	0.540274	0.532068	0.536139	RFE(estimator=LogisticRegression(), n_features_to_select=4)
SelfTrainingClassifier	0.693519	0.540274	0.532068	0.536139	RFE(estimator=LogisticRegression(), n_features_to_select=4)
KNeighborsClassifier	0.69432	0.539345	0.535021	0.537174	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
SelfTrainingClassifier	0.69432	0.539345	0.535021	0.537174	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
KNeighborsClassifier	0.69432	0.539345	0.535021	0.537174	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
SelfTrainingClassifier	0.69432	0.539345	0.535021	0.537174	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
KNeighborsClassifier	0.69432	0.539345	0.535021	0.537174	RFECV(estimator=SVC(), min_features_to_select=4)
SelfTrainingClassifier	0.69432	0.539345	0.535021	0.537174	RFECV(estimator=SVC(), min_features_to_select=4)
KNeighborsClassifier	0.69432	0.539345	0.535021	0.537174	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
SelfTrainingClassifier	0.69432	0.539345	0.535021	0.537174	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
KNeighborsClassifier	0.694286	0.53923	0.535021	0.537117	SelectKBest(mutual_info_classif, k=4)
SelfTrainingClassifier	0.694286	0.53923	0.535021	0.537117	SelectKBest(mutual_info_classif, k=4)
KNeighborsClassifier	0.687752	0.537017	0.518776	0.527739	SparsePCA(n_components=4)
SelfTrainingClassifier	0.687752	0.537017	0.518776	0.527739	SparsePCA(n_components=4)
SGDClassifier	0.695259	0.53659	0.539873	0.538227	SelectKBest(f_classif, k=4)
RandomForestClassifier	0.682939	0.536465	0.505907	0.520738	PCA(n_components=4)
RandomForestClassifier	0.680885	0.535392	0.501055	0.517655	SparsePCA(n_components=4)
KNeighborsClassifier	0.692997	0.533124	0.536498	0.534805	PCA(n_components=4)
SelfTrainingClassifier	0.692997	0.533124	0.536498	0.534805	PCA(n_components=4)

AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	SelectKBest(f_classif, k=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	SelectKBest(mutual_info_classif, k=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	SelectKBest(f_regression, k=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	RFE(estimator=LogisticRegression(), n_features_to_select=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	RFE(estimator=SVC(), n_features_to_select=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	RFE(estimator=MLPClassifier(), n_features_to_select=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	RFECV(estimator=SVC(), min_features_to_select=4)
AdaBoostClassifier	0.695745	0.52735	0.549156	0.538032	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
SGDClassifier	0.723537	0.524789	0.629747	0.572497	RFE(estimator=MLPClassifier(), n_features_to_select=4)
BaggingClassifier	0.67295	0.522297	0.489241	0.505229	RFECV(estimator=SVC(), min_features_to_select=4)
RandomForestClassifier	0.6837	0.521115	0.520675	0.520895	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
RandomForestClassifier	0.675248	0.519709	0.49789	0.508566	RFECV(estimator=SVC(), min_features_to_select=4)
RandomForestClassifier	0.676525	0.518269	0.502743	0.510388	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
RandomForestClassifier	0.678684	0.517574	0.509494	0.513502	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	SelectKBest(f_classif, k=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	SelectKBest(mutual_info_classif, k=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	SelectKBest(f_regression, k=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	RFE(estimator=SVC(), n_features_to_select=4)
LGBMClassifier	0.66674	0.517202	0.475738	0.495604	RFE(estimator=MLPClassifier(), n_features_to_select=4)

LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	RFECV(estimator=SVC(), min_features_to_select=4)
LGBMClassifier	0.666 74	0.51 7202	0.47 5738	0.49 5604	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
RandomForestClassifier	0.686 027	0.51 7163	0.53 0802	0.52 3894	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
RandomForestClassifier	0.674 761	0.51 6827	0.49 8945	0.50 7729	SelectKBest(f_regression, k=4)
SGDClassifier	0.595 329	0.51 64	0.27 2363	0.35 663	MiniBatchSparsePCA(n_components=4)
RandomForestClassifier	0.675 242	0.51 5968	0.50 1055	0.50 8402	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
BaggingClassifier	0.676 608	0.51 5608	0.50 5274	0.51 0389	MiniBatchSparsePCA(n_components=4)
LGBMClassifier	0.675 714	0.51 5578	0.50 2743	0.50 9079	SparsePCA(n_components=4)
PassiveAggressiveClassifier	0.599 883	0.51 5405	0.28 5865	0.36 7757	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
RandomForestClassifier	0.677 186	0.51 5327	0.50 7173	0.51 1217	SelectKBest(f_classif, k=4)
RandomForestClassifier	0.676 916	0.51 4445	0.50 7173	0.51 0783	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
RandomForestClassifier	0.680 588	0.51 3254	0.51 8776	0.51 6	MiniBatchSparsePCA(n_components=4)
BaggingClassifier	0.677 674	0.51 2149	0.51 1392	0.51 177	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
RandomForestClassifier	0.681 039	0.51 2129	0.52 1097	0.51 6574	SelectKBest(mutual_info_classif, k=4)
RandomForestClassifier	0.679 896	0.51 1992	0.51 7932	0.51 4945	RFE(estimator=MLPClassifier(), n_features_to_select=4)
BaggingClassifier	0.666 817	0.51 0629	0.48 1435	0.49 5602	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
RandomForestClassifier	0.670 333	0.50 8254	0.49 3671	0.50 0856	RFE(estimator=LogisticRegression(), n_features_to_select=4)
LGBMClassifier	0.674 685	0.50 728	0.50 7173	0.50 7227	PCA(n_components=4)
HistGradientBoostingClassifier	0.673 884	0.50 6121	0.50 5907	0.50 6014	SparsePCA(n_components=4)
HistGradientBoostingClassifier	0.672 352	0.50 5632	0.50 1899	0.50 3759	MiniBatchSparsePCA(n_components=4)
HistGradientBoostingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	SelectKBest(f_classif, k=4)
HistGradientBoostingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	SelectKBest(mutual_info_classif, k=4)
HistGradientBoostingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	SelectKBest(f_regression, k=4)
HistGradientBoostingClassifier	0.668 121	0.50 3896	0.49 1139	0.49 7436	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)

HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	RFE(estimator=LogisticRegression(), n_features_to_select=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	RFE(estimator=SVC(), n_features_to_select=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	RFE(estimator=MLPClassifier(), n_features_to_select=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	RFECV(estimator=SVC(), min_features_to_select=4)
HistGradientBoostingClassifier	0.668121	0.503896	0.491139	0.497436	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
BaggingClassifier	0.675725	0.503515	0.513713	0.508563	RFE(estimator=LogisticRegression(), n_features_to_select=4)
BaggingClassifier	0.6546	0.502576	0.452743	0.47636	RFE(estimator=SVC(), n_features_to_select=4)
HistGradientBoostingClassifier	0.675872	0.501538	0.516034	0.508683	PCA(n_components=4)
GradientBoostingClassifier	0.671435	0.501155	0.503376	0.502263	MiniBatchSparsePCA(n_components=4)
ExtraTreeClassifier	0.662907	0.500994	0.478481	0.489479	RFE(estimator=LogisticRegression(), n_features_to_select=4)
BaggingClassifier	0.664776	0.500764	0.484177	0.492331	SelectKBest(f_regression, k=4)
SGDClassifier	0.701601	0.500178	0.593038	0.542664	SelectKBest(mutual_info_classif, k=4)
XGBClassifier	0.659507	0.49877	0.470464	0.484204	PCA(n_components=4)
BaggingClassifier	0.6665	0.498715	0.491139	0.494898	SelectKBest(mutual_info_classif, k=4)
BaggingClassifier	0.656272	0.497276	0.462236	0.479117	SelectKBest(f_classif, k=4)
SGDClassifier	0.711569	0.497151	0.625949	0.554165	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
XGBClassifier	0.659435	0.49689	0.471941	0.484094	SparsePCA(n_components=4)
LGBMClassifier	0.668868	0.495004	0.501688	0.498324	MiniBatchSparsePCA(n_components=4)
BaggingClassifier	0.649383	0.493452	0.445148	0.468057	PCA(n_components=4)
RandomForestClassifier	0.66591	0.492246	0.49557	0.493902	RFE(estimator=SVC(), n_features_to_select=4)
XGBClassifier	0.664648	0.490764	0.493249	0.492003	MiniBatchSparsePCA(n_components=4)
SGDClassifier	0.717709	0.490028	0.653165	0.559957	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
BaggingClassifier	0.668409	0.489388	0.505907	0.49751	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)

BaggingClassifier	0.653159	0.487316	0.462025	0.474334	RFE(estimator=MLPClassifier(), n_features_to_select=4)
PassiveAggressiveClassifier	0.669294	0.486649	0.511392	0.498714	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	SelectKBest(f_classif, k=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	SelectKBest(mutual_info_classif, k=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	SelectKBest(f_regression, k=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	RFE(estimator=LogisticRegression(), n_features_to_select=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	RFE(estimator=SVC(), n_features_to_select=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	RFE(estimator=MLPClassifier(), n_features_to_select=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	RFECV(estimator=SVC(), min_features_to_select=4)
XGBClassifier	0.66076	0.482637	0.489662	0.486124	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
PassiveAggressiveClassifier	0.690191	0.482274	0.579747	0.526538	RFE(estimator=LogisticRegression(), n_features_to_select=4)
BaggingClassifier	0.65093	0.481997	0.460338	0.470918	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
BaggingClassifier	0.655728	0.481023	0.475949	0.478473	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
BaggingClassifier	0.649005	0.479416	0.456962	0.46792	SparsePCA(n_components=4)
AdaBoostClassifier	0.645028	0.478793	0.445359	0.461471	MiniBatchSparsePCA(n_components=4)
BaggingClassifier	0.660555	0.477208	0.494726	0.485809	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
SGDClassifier	0.578616	0.471423	0.245359	0.322742	RFECV(estimator=SVC(), min_features_to_select=4)
SGDClassifier	0.689922	0.465653	0.600633	0.524599	SparsePCA(n_components=4)
PassiveAggressiveClassifier	0.736446	0.464739	0.749367	0.57369	RFECV(estimator=SVC(), min_features_to_select=4)
SGDClassifier	0.699235	0.462627	0.634599	0.535136	PCA(n_components=4)
DecisionTreeClassifier	0.655117	0.459086	0.498312	0.477896	MiniBatchSparsePCA(n_components=4)
ExtraTreeClassifier	0.645447	0.457938	0.468565	0.463191	RFECV(estimator=SVC(), min_features_to_select=4)

DecisionTreeClassifier	0.65493	0.457906	0.499156	0.477642	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)
DecisionTreeClassifier	0.650111	0.457445	0.484177	0.470431	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
ExtraTreeClassifier	0.635472	0.454209	0.440506	0.447253	RFECV(estimator=MLPClassifier(), min_features_to_select=4)
DecisionTreeClassifier	0.651466	0.453786	0.493038	0.472599	SelectKBest(f_regression, k=4)
DecisionTreeClassifier	0.652971	0.452823	0.499156	0.474862	RFE(estimator=LogisticRegression(), n_features_to_select=4)
DecisionTreeClassifier	0.65225	0.452798	0.496835	0.473795	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
ExtraTreeClassifier	0.659253	0.452692	0.519831	0.483944	PCA(n_components=4)
DecisionTreeClassifier	0.658161	0.45175	0.517511	0.482399	RFE(estimator=MLPClassifier(), n_features_to_select=4)
DecisionTreeClassifier	0.651089	0.451638	0.494515	0.472105	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
DecisionTreeClassifier	0.642554	0.450865	0.467511	0.459037	PCA(n_components=4)
DecisionTreeClassifier	0.650484	0.448618	0.496414	0.471307	RFECV(estimator=SVC(), min_features_to_select=4)
PassiveAggressiveClassifier	0.665584	0.448485	0.546414	0.49263	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
DecisionTreeClassifier	0.654435	0.44811	0.510127	0.477111	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
DecisionTreeClassifier	0.655417	0.447857	0.513713	0.47853	RFE(estimator=SVC(), n_features_to_select=4)
ExtraTreeClassifier	0.630028	0.446238	0.431646	0.43882	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
ExtraTreeClassifier	0.636294	0.445643	0.453165	0.449372	SelectKBest(mutual_info_classif, k=4)
DecisionTreeClassifier	0.654149	0.444181	0.514557	0.476786	SelectKBest(f_classif, k=4)
ExtraTreeClassifier	0.624946	0.44417	0.417089	0.430203	SparsePCA(n_components=4)
DecisionTreeClassifier	0.649385	0.443965	0.498945	0.469852	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
DecisionTreeClassifier	0.647917	0.443457	0.494726	0.46769	SelectKBest(mutual_info_classif, k=4)
SGDClassifier	0.658097	0.440341	0.533333	0.482397	RFE(estimator=LogisticRegression(), n_features_to_select=4)
ExtraTreeClassifier	0.650133	0.435764	0.51308	0.471272	RFE(estimator=RandomForestClassifier(), n_features_to_select=4)
ExtraTreeClassifier	0.650857	0.43348	0.518987	0.472396	MiniBatchSparsePCA(n_components=4)
DecisionTreeClassifier	0.629059	0.433244	0.444304	0.438704	SparsePCA(n_components=4)
ExtraTreeClassifier	0.639881	0.426428	0.491561	0.456684	SelectKBest(f_classif, k=4)
ExtraTreeClassifier	0.62958	0.425152	0.457173	0.440581	RFE(estimator=MLPClassifier(), n_features_to_select=4)
ExtraTreeClassifier	0.63923	0.42411	0.492827	0.455894	RFE(estimator=DecisionTreeClassifier(), n_features_to_select=4)

ExtraTreeClassifier	0.623144	0.421853	0.43903	0.43027	SelectKBest(f_regression, k=4)
ExtraTreeClassifier	0.629722	0.418254	0.467932	0.441701	RFECV(estimator=DecisionTreeClassifier(), min_features_to_select=4)
ExtraTreeClassifier	0.616138	0.41791	0.419409	0.418659	RFECV(estimator=RandomForestClassifier(), min_features_to_select=4)
ExtraTreeClassifier	0.611507	0.414608	0.407173	0.410857	RFE(estimator=SVC(), n_features_to_select=4)
PassiveAggressiveClassifier	0.682597	0.414557	0.666878	0.511282	MiniBatchSparsePCA(n_components=4)
PassiveAggressiveClassifier	0.633 97	0.402561	0.510759	0.450251	PCA(n_components=4)
SGDClassifier	0.715404	0.386526	0.876371	0.53645	RFECV(estimator=LogisticRegression(), min_features_to_select=4)
PassiveAggressiveClassifier	0.505663	0.355556	0.027004	0.050196	SelectKBest(f_regression, k=4)
PassiveAggressiveClassifier	0.616375	0.346182	0.589241	0.436134	SparsePCA(n_components=4)
BernoulliNB	0.536973	0.33288	0.20654	0.254915	PCA(n_components=4)
BernoulliNB	0.536973	0.33288	0.20654	0.254915	SparsePCA(n_components=4)
BernoulliNB	0.530646	0.301609	0.237342	0.265643	MiniBatchSparsePCA(n_components=4)
SGDClassifier	0.549 54	0.291753	0.44557	0.352617	SelectKBest(f_regression, k=4)
SGD Classifier	0.503 94	0.259615	0.091139	0.134916	RFE(estimator=SVC(), n_features_to_select=4)