

FRAUD APP DETECTION USING MACHINE LEARNING

Ms. S Manjula, K. Aashritha Reddy, K. Akhila, G. Akshitha

¹Assoc. Professor & HOD, Department of ECE, Bhoj Reddy Engineering College for Women, Hyderabad, India

^{2,3}B.Tech Students, Department of ECE, Bhoj Reddy Engineering College for Women, Hyderabad, India

Abstract: Due to the rise in online transactions and fraud, strong solutions are needed to protect financial transactions and sensitive data. A new supervised machine learning method classifies dangerous and benign network fraud applications. Supervised learning and feature selection were utilized to develop the optimum detection success rate model. This research demonstrated that Random Forest-based machine learning with wrapper feature selection classifies network fraud applications better than SVM. The dataset is used to identify network fraud applications using SVM and RANDOM FOREST supervised machine learning to assess performance. Comparative analysis demonstrates that our model outperforms other methods in Fraud Application detection.

Rule-based fraud detection systems fail to adapt to changing fraud methods and trends. High false positive rates and frequent manual upgrades characterize these systems. They cannot identify new fraud trends, leaving security weaknesses.

We propose a Machine Learning-based Fraud App to address these constraints. The software can understand complicated fraud patterns by studying past data and using advanced ML algorithms, enhancing accuracy and flexibility. This allows real-time fraud detection with fewer false positives, improving user experience by reducing security alarms. Adaptive learning, continual improvement, and fraud pattern detection are benefits of the suggested system. The app's powerful ML models boost fraud detection accuracy and keep up with new threats, improving security, fraud losses, and online transaction confidence for financial institutions and customers.

I. INTRODUCTION

The danger of fraudulent activities has increased in today's quickly evolving technology world, where digital transactions and online activities have become a crucial aspect of our daily lives. The rise of fraudulent apps, which pose a hazard to both consumers and organizations, is one area that has attracted a lot of attention. These phony applications aim to trick users into disclosing private information, engaging in harmful activity, or resulting in financial loss. It is therefore critical to have reliable and fast ways to identify and stop these fake app instances.

The development of Machine Learning (ML) approaches has fundamentally changed how we tackle difficult issues, such as fraud detection. We can develop intelligent systems that can identify patterns, anomalies, and hidden links in massive datasets by using the power of machine learning algorithms. Using machine learning, the "Fraud App Detection using Machine Learning" initiative seeks to detect and reduce the risks related to fraudulent mobile apps.

Our lives are now more convenient than ever because to the growing dependence on mobile apps for a variety of functions, including communication and financial transactions. On the other hand, as technology has advanced, so too have fraudulent mobile apps proliferated, preying on user confidence and jeopardizing confidential data.

These fraudulent applications have the potential to cause identity theft, money losses, and a decline in trust for digital networks.

In order to build a safer digital environment, this problem must be resolved. We are able to proactively detect and prevent fake app instances by using machine learning. The need to create a proactive defense system that protects users from fraudulent applications, fosters user confidence in the app ecosystem, and eventually guarantees a safe and smooth digital experience for everyone is what drives this project.

Conceptual Framework

The novel idea was called Hidden Naïve Bayes, which has more advantages than regular Naïve Bayes. It analyzes a lot of network data and takes into account the intricate characteristics of attack behaviors to increase the efficiency of detection speed and accuracy.

II. METHODOLOGY

Here are some Python-related facts. At the moment, Python is the most popular high-level, multipurpose programming language. Python programming supports both procedural and object-oriented paradigms. Compared to other programming languages like Java, Python applications are often smaller. Because of the language's indentation requirements, programmers type comparatively less, which makes their work consistently readable. Nearly all of the major internet businesses, including Google, Amazon, Facebook, Instagram, Dropbox, and Uber, employ the Python programming language. Python's largest asset is its vast library of standard libraries, which may be used for the following: • Machine Learning

- GUI Applications (like Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Steps in Python Development

In February 1991, Guido Van Rossum released version 0.9.0 of the Python code at alt. sources. Exception handling, functions, and the fundamental data types list, dict, and str were previously provided in this version. It included a module structure and was object oriented as well. January 1994 saw the introduction of Python version 1.0. The functional programming tools lambda, map, filter, and reduce—which Guido Van Rossum never liked—were the main new features added to this edition. October 2000, six and a half years later, saw the release of Python 2.0. List comprehensions, a comprehensive garbage collector, and support for Unicode were all implemented in this edition. Python had further success in versions 2.x for a further 8 years, until the next major release, Python 3.0 (sometimes referred to as "Python 3000" and "Py3K"), was published. Python 2.x and Python 3 are not backwards compatible. The goal of Python 3 was to eliminate redundant programming techniques and modules, which partially or fully complies with the 13th rule of the Zen of Python, which states that "there should be one -- and preferably only one -- obvious way to do it." A few changes to Python 7.3:

III. SYSTEM DESIGN

In fraud detection applications, system design include drafting a thorough implementation strategy for the architecture. It entails defining data flows, algorithms, and decision-making procedures in addition to describing the modules, components, and how they interact. Considerations such as data sources, feature extraction, machine learning models, rule engines, and alert systems are part of a well-designed system. The objective is to provide a unified and effective system that can quickly detect and react to fraudulent activity, offering a strong defense against different kinds of fraud inside the application.

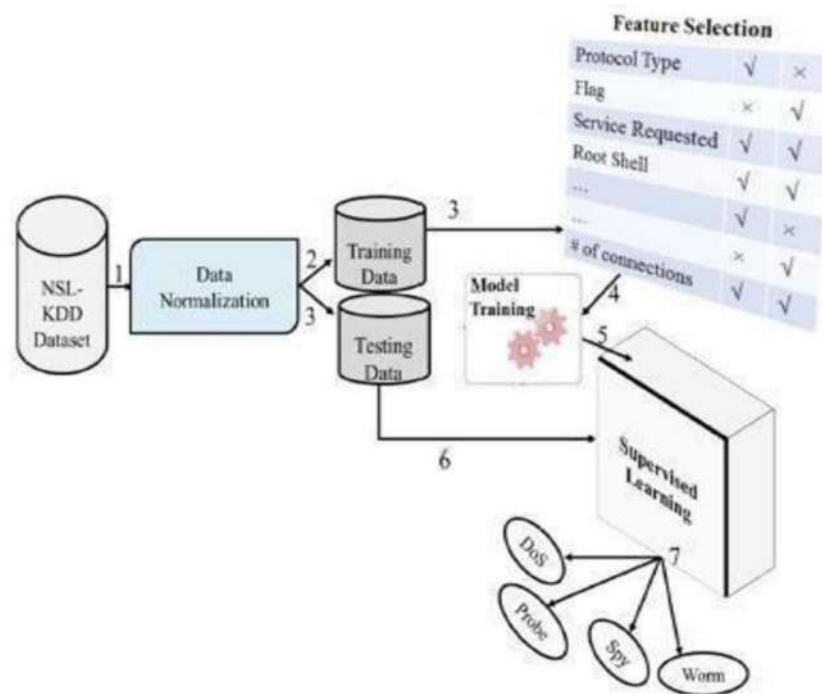


Fig Proposed Supervised machine learning classifier System

In order to minimize the dimensionality of data, feature selection is a crucial component of machine learning, and a robust feature selection technique has been the subject of much study. Both the filter approach and the wrapper method have been utilized for feature selection. The filter technique selects features based on how well they perform in a variety of statistical tests that gauge a feature's importance by examining how well it correlates with an outcome or dependent variable. By calculating a subset of features' utility with respect to the dependent variable, the wrapper technique determines a subset of features. As a result, filter techniques operate independently of machine learning algorithms, while wrapper approaches rely on the model's training machine learning process to determine which feature subset is optimal. Using a classification technique, a subset evaluator in the wrapper approach employs every feasible subset to persuade classifiers based on the characteristics of each subset. The classifier evaluates the subset of features that the classification algorithm performs best on before determining the method's predicted accuracy. The classification algorithm uses different weights or ranks than the ranker methods. While the filter technique works well for data mining tests due to its millions of features, the wrapper method is better suited for machine learning tests

IV. .IMPLEMENTATION

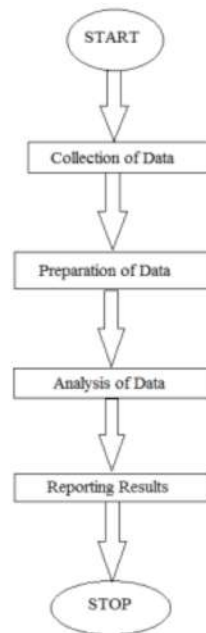


Fig Steps followed for analysis

The steps that are followed for analysis of data is shown in fig are:

Collection of data: The first step is to collect the data. The data can be collected from various methods such as crawling, Application Program Interface (API) or directly from the company. In this research work, the data is collected directly from the company itself.

Preparation of data: After collecting data, the data is prepared for performing analysis efficiently. The obtained dataset is composed of various attributes that are not necessary for analysis, therefore it is better to prepare the data according to the need so that the algorithm generates accurate results.

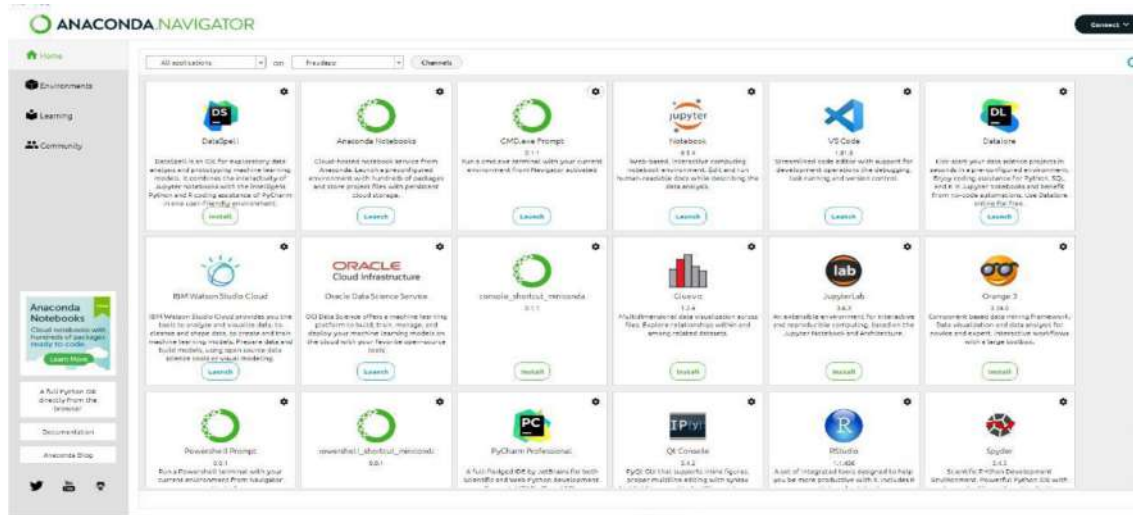
Analysis of data: After preparing the data, the analysis is performed using various algorithms according to the need. In this work, random forest algorithm is used. It is implemented using caret and randomForest package. Caret package consist of set of functions for training and creating classification and regression predictive models. It is composed of various tools such as splitting of data, Pre-processing of data, Selection of features from the data, Tuning of model using resampling method and Estimation of variable importance. randomForest package implements Random Forest Algorithm that was introduced by Brieman in 2001. This is used for the classification and regression of data.

Reporting results: After the complete analysis of data, results are obtained. The results are produced by calling the randomForest variable. It has been observed that the random forest has performed classification and results are displayed .

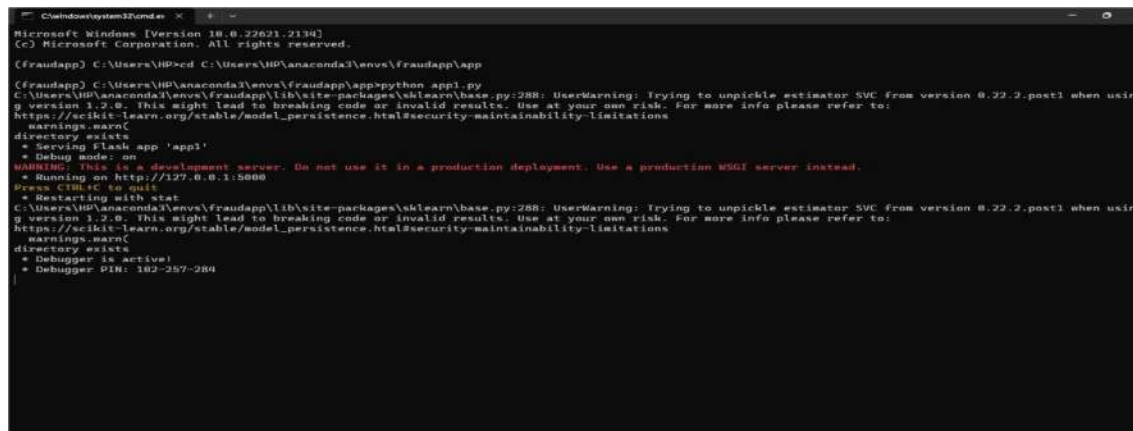
V. RESULTS

The results of simulation are shown for fraud app using machine learning algorithm.

Step 1: Open Anaconda Navigator and launch CMD.exe Prompt



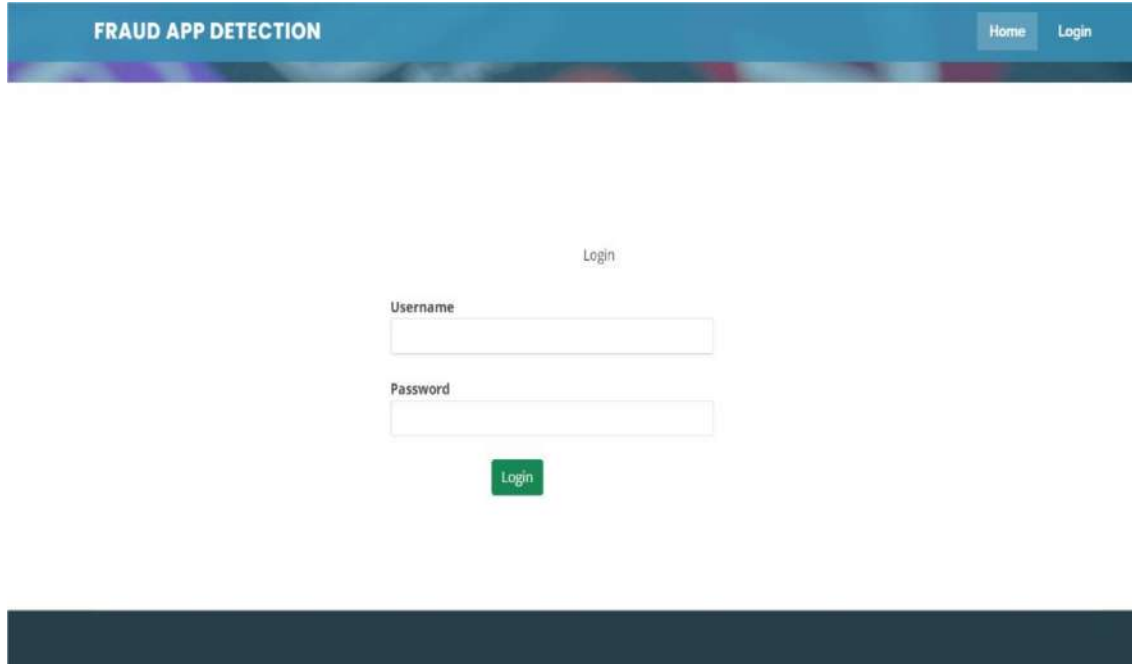
Step 2: Execute the Python code and get the URL link



Step 3: Paste the URL link on browser to get the Web Page

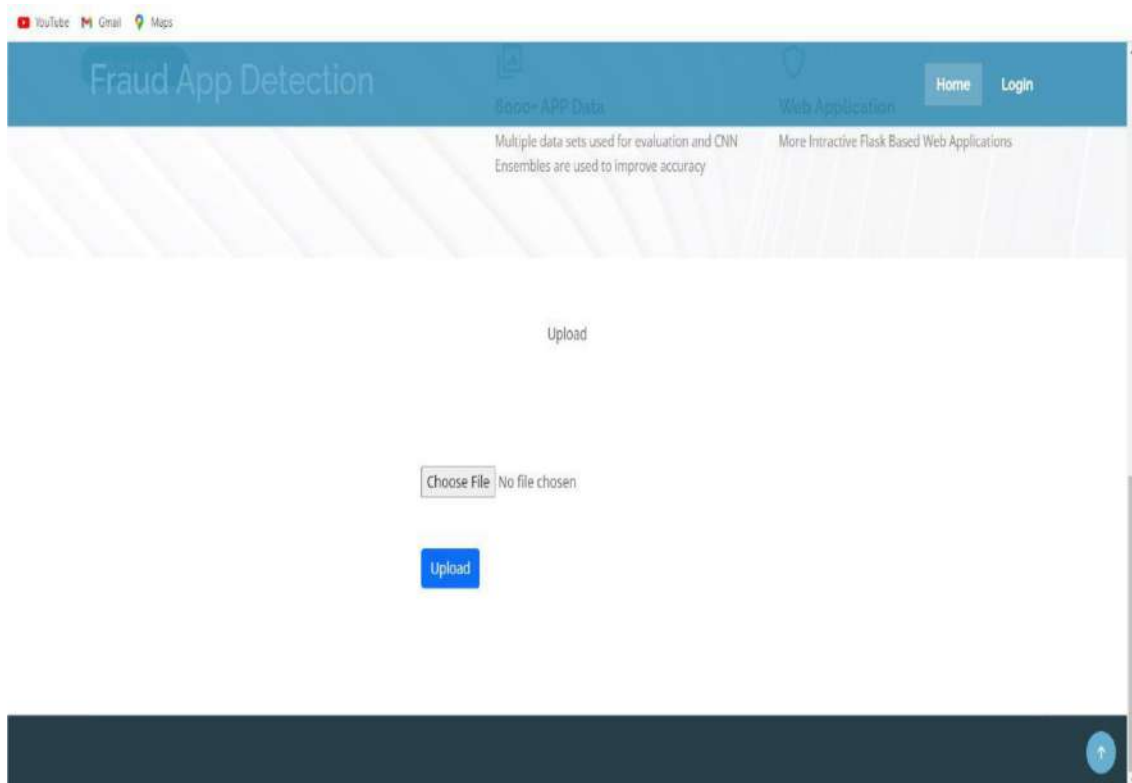


Step 4: Enter the Login credentials which are already Predefined.



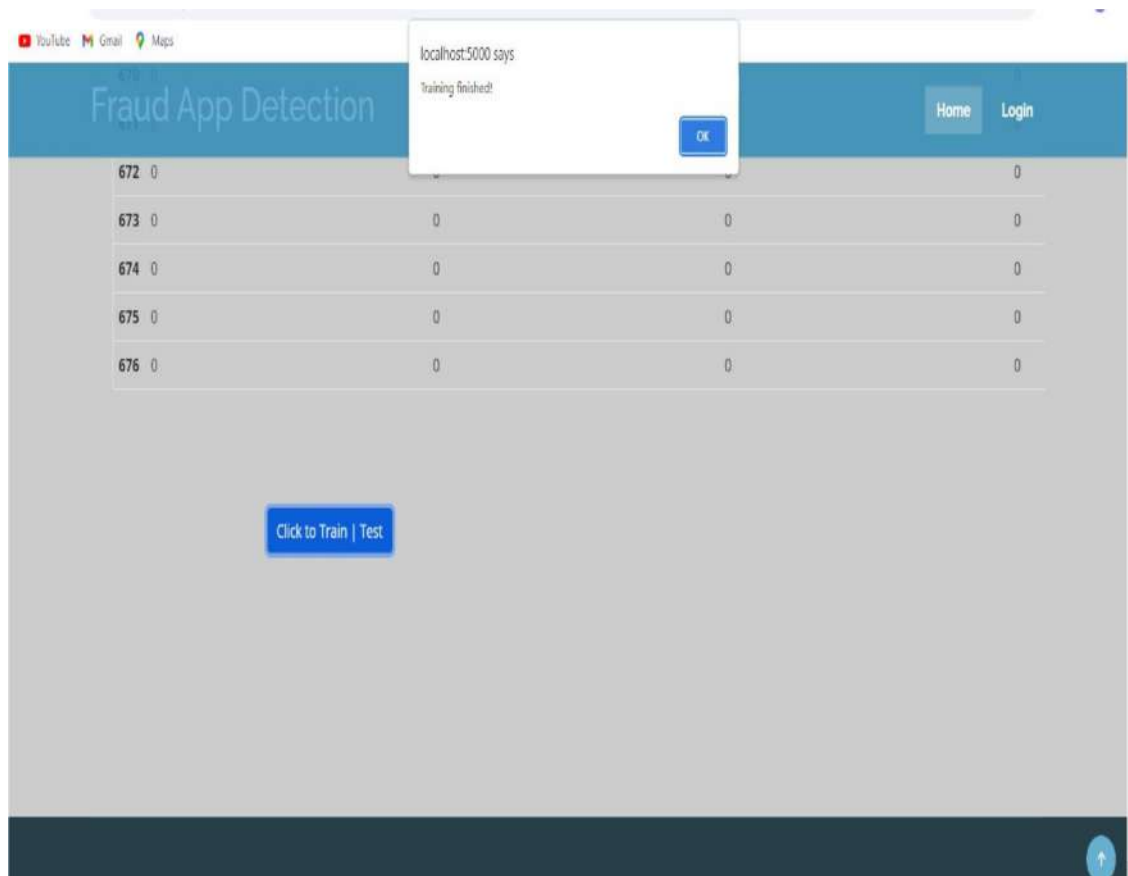
The screenshot shows the login interface of the 'Fraud App Detection' web application. At the top, there is a blue header bar with the title 'FRAUD APP DETECTION' on the left and 'Home' and 'Login' buttons on the right. Below the header, the word 'Login' is centered. Underneath, there are two input fields: 'Username' and 'Password'. A green 'Login' button is positioned below the password field. The background of the page is a light blue gradient with a subtle pattern of white lines.

Step 5: Upload the data Set for testing.

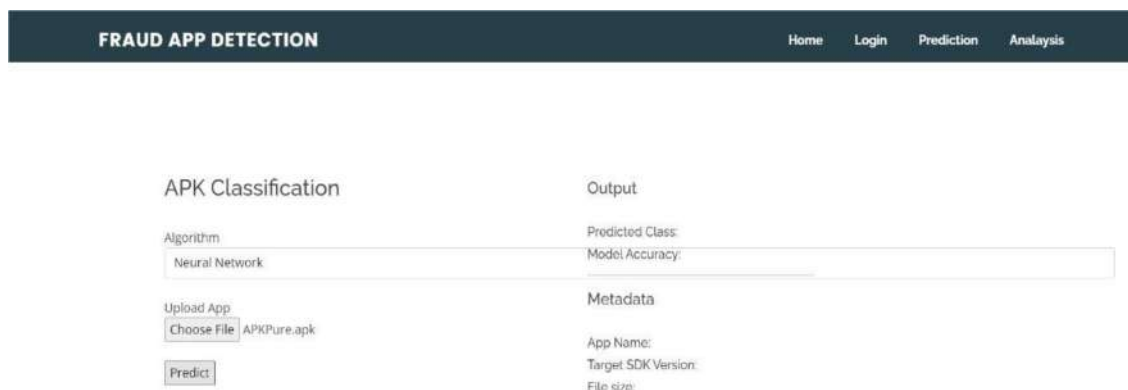


The screenshot shows the upload interface of the 'Fraud App Detection' web application. At the top, there is a blue header bar with the title 'Fraud App Detection' on the left and 'Home' and 'Login' buttons on the right. Below the header, there are two sections: 'Isoco- APP Data' and 'Web Application'. The 'Isoco- APP Data' section contains the text 'Multiple data sets used for evaluation and CNN Ensembles are used to improve accuracy'. The 'Web Application' section contains the text 'More Interactive Flask Based Web Applications'. Below these sections, the word 'Upload' is centered. Underneath, there is a 'Choose File' button and a 'No file chosen' text. A blue 'Upload' button is positioned below the 'Choose File' button. The background of the page is a light blue gradient with a subtle pattern of white lines.

Step 6: Click to train the Dataset.



Step 7: apk file of any app can be chosen and uploaded to get the status of app.



Final Output: The different features and status of app is verified here.

FRAUD APP DETECTION		Home	Login	Prediction	Analysis
APK Classification		Output			
Algorithm		Predicted Class: Malware			
Neural Network		Model Accuracy: 92.26 %			
Upload App		Metadata			
Choose File SVG.apk		App Name: SVG Viewer			
Predict		Target SDK Version: 29			
		File size: 0.1 MB			

VI. CONCLUSION AND FUTURE SCOPE

Feature selection is an essential part of machine learning to reduce the dimensionality of data, and a robust feature selection method has been extensively researched. For feature selection, both the wrapper technique and the filter approach have been used. Using a range of statistical tests that measure a feature's significance by looking at how well it connects with an outcome or dependent variable, the filter approach chooses characteristics. The wrapper approach selects a subset of features by calculating the utility of a subset of features with regard to the dependent variable. Therefore, whereas wrapper methods rely on the model's training machine learning process to identify which feature subset is ideal, filter strategies work independently of machine learning algorithms. In the wrapper method, a subset evaluator uses a classification strategy to use all possible subsets to convince classifiers based on the attributes of each subset. Prior to estimating the expected accuracy of the approach, the classifier assesses the subset of characteristics on which the classification algorithm works optimally. In contrast to ranker approaches, the classification algorithm employs distinct weights or ranks. The wrapper approach is more appropriate for machine learning testing, while the filter strategy, with its millions of features, works well for data mining tests.

References

- [1] H. Song, M. J. Lynch, and J. K. Cochran, "A macro-social exploratory analysis of the rate of interstate cyber-victimization," *American Journal of Criminal Justice*, vol. 41, no. 3, pp. 583–601, 2016.
- [2] P. Alaei and F. Noorbehbahani, "Incremental anomaly-based Fraud Application detection system using limited labeled data," in *Web Research (ICWR), 2017 3th International Conference on*, 2017, pp. 178–184.
- [3] M. Saber, S. Chadli, M. Emharraf, and I. El Farissi, "Modeling and implementation approach to evaluate the Fraud Application detection system," in *International Conference on Networked Systems*, 2015, pp. 513–517.
- [4] M. Tavallaei, N. Stakhonova, and A. A. Ghorbani, "Toward credible evaluation of anomaly-based Fraud Application-detection methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 516–524, 2010.

- [5] Gavni Ranjitha, Boyalu Bhavani, Pudutha Akhila, Detection Of Hyperspectral Images, International Journal of Multidisciplinary Engineering in Current Research - IJMEC Volume 8, Issue 5, May-2023, <http://ijmec.com/>, ISSN: 2456-4265.
- [6] Vishnuvardhan Reddy, G. Shivani, J. Sowmya, M. Jyothi, P. Sai Prasad Reddy, B. Vinay Kumar, Design And Analysis Of Auditorium By Using Staad PRO, International Journal of Multidisciplinary Engineering in Current Research - IJMEC Volume 8, Issue 5, May-2023, <http://ijmec.com/>, ISSN: 2456-4265.
- [7] A. S. Ashoor and S. Gore, "Importance of Fraud Application detection system (IDS)," International Journal of Scientific and Engineering Research, vol. 2, no. 1, pp. 1–4, 2011.
- [8] M. Zamani and M. Movahedi, "Machine learning techniques for Fraud Application detection," arXiv preprint arXiv:1312.2177, 2013.
- [9] N. Chakraborty, "Fraud Application detection system and Fraud Application prevention system: A comparative study," International Journal of Computing and Business Research (IJCBR) ISSN (Online), pp. 2229– 6166, 2013.