**Prediction of the network attacks by finding the best accuracy using supervised machine learning algorithm**

**Abstract:**

Generally, to create data for the Intrusion Detection System (IDS), it is necessary to set the real working environment to explore all the possibilities of attacks, which is expensive. Software to detect network intrusions protects a computer network from unauthorized users, including perhaps insiders. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between "bad" connections, called intrusions or attacks, and "good" normal connections. To prevent this problem in network sectors have to predict whether the connection is attacked or not from KDDCup99 dataset using machine learning techniques. The aim is to investigate machine learning based techniques for better packet connection transfers forecasting by prediction results in best accuracy. To propose a machine learning-based method to accurately predict the DOS, R2L, UU2R, Probe and overall attacks by prediction results in the form of best accuracy from comparing supervise classification machine learning algorithms. Additionally, to compare and discuss the performance of various machine learning algorithms from the given dataset with evaluation classification report, identify the confusion matrix and to categorizing data from priority and the result shows that the effectiveness of the proposed machine learning algorithm technique can be compared with best accuracy with precision, Recall and F1 Score.

**Keywords:** dataset, Machine learning-Classification method, python, Prediction of Accuracy result.

**INTRODUCTION**

**Domain overview**

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

Data scientists use many different kinds of machine learning algorithms to discover patterns in python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they “learn” about data to make predictions: supervised and unsupervised learning. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function from input variables(X) to discrete output variables(y). In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.

Analyses Predicts

Machine Learning

Past Dataset

Trains

Fig: Process of Machine learning

[Supervised Machine Learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning/) **is the** majority of practical machine learning uses supervised learning. Supervised learning is where have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output**is y = f(X).** The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include **logistic regression**, **multi-class classification**, **Decision Trees** and **support vector machines etc**. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. Supervised learning problems can be further grouped into **Classification** problems. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for categorical for classification. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. A classification problem is when the output variable is a category, such as “red” or “blue”.

**C**OMMUNICATIONS based on radio wave propagation, which cannot be confined and emit in all directions, can be the victim of various cyberattacks. The main consequence of this “wild propagation” of radio waves is that unauthorized persons may listen to the network communications and possibly from outside a building. There are multiple risks of the poor protection of a wireless network, including, data interception, diversion of connection for illicit access to a local network, jamming signals, or dummy commands for the denials of service, etc. Different approaches can be studied to protect these wireless communications from such attacks. In this paper, we consider that the first step to countermeasure the attacks consists in detecting them and recognizing the type of attack in order to adapt the action to involve. Intrusion detection systems (IDSs) can detect an abnormal activity on an analyzed target. There are three major IDS families: the network intrusion detection system (NIDS) that monitors the security state at the network level, the host-based intrusion detection system (HIDS) that monitors the security state at the host level, and an hybrid IDS that combines NIDS and HIDS. The major difference between NIDS and HIDS is that the HIDS is particularly effective in determining whether a host is contaminated, whereas an NIDS can monitor an entire network. However, the IDS mainly works on the upper layers of the open system interconnection (OSI) model and do not protect the wireless communication links. Moreover, the intrusion detection principles used need to be deployed on all terminals. In this study, we work on a solution that outsources the attack detection function by analyzing the wireless electromagnetic (EM) activity. It consists in taking data from antennas and receivers

perfectly independent from the protected communication networks, and then, applying classification algorithms on the data.

The detection of jamming attacks on wireless link networks was studied by analyzing the received signal strength indication (RSSI) received by a station. The RSSI is used by the IEEE 802.11 standard to measure the relative quality of the received signal. The detection is based on a synchronization indicator together with an adaptive signal to noise plus jammer power ratio. In our case, as the monitoring solution is outsourced, we are not limited to the indicators of the standard. We selected the EM spectra collected by an independent bench, since they provide more information than the RSSI for the monitoring of the physical link.

**Problem Description**

Lately, an internet network company in Japan has been facing huge losses due to malicious server attacks. They've encountered breach in data security, reduced data transfer speed and intermittent breakdowns in user-user & user-network connections. When asked, a company official said, “there’s a significant dip in the number of active users on our network ". The company is looking are some predictive analytics solution to help them understand, detect and counter the attacks and make their network connection secure. Think of a connection as a sequence of TCP packets starting and ending at some well-defined times, between which data flows to and from a source IP address to a target IP address under some well-defined protocol. In total, there are 3 major type of attacks to which their network is vulnerable to. But, 3 of them cause the maximum damage. In this challenge, you are given an anonymised sample dataset of server connections. You have to predict the type of attack(s) like Dos, R2L, U2R, Probe.

**Scope:**

The scope of this project is to investigate a dataset of network connection attacks for KDD records for medical sector using machine learning technique. To identifying network connection is attacked or not.

**EXISTING SYSTEM**

To focuses on the conception of a monitoring system able to detect and classify jamming and protocol-based attacks and achieve this goal, we proposed to outsource the attack detection function from the network to protect and used an antenna to monitor the spectrum over the time. The Wi-Fi network and the attacks were carried out in an anechoic chamber to avoid disturbing other Wi-Fi communication networks in the vicinity. The spectra highlights that the frequencies of interest belong to the communication channel between 2.402 and 2.422 GHz. Focusing the analysis on this 20-MHz frequency band permits to construct a classification model to overcome the problems induced by the utilization of the adjacent channels that can be or not occupied by other Wi-Fi communications. On these frequencies, the proposed estimation model shows good results in the prediction of attacks. In addition, the correction using the K spectra nearest in time permits to correct most of the miss classification.

The development of connected devices and their daily use is presently at the origin of the omnipresence of Wi-Fi wireless networks. However, these Wi-Fi networks are often vulnerable, and can be used by malicious people to disturb services, intercept sensitive data, or to gain access to the system. In railways, trains are now equipped with wireless communication systems for operational purposes or for passenger services. In both cases, defense strategies have to be developed to prevent the misuses of the networks. The first objective of this study is to propose a monitoring solution, which is independent of the communication networks, to detect the occurrence of attacks. The second objective is to develop a method that is able to classify attacks of different types: the intentional electromagnetic interference, i.e., jamming attacks and the protocol-based attacks. To perform these analyses, we propose to monitor and to analyze electromagnetic (EM) signals received by a monitoring antenna and a receiver collecting the EM spectra. After that, we build a classification protocol following two steps: the first consists in the construction of a support vector machine (SVM) classification model using the collected spectra, and the second step uses this SVM model to predict the class of the attack (if any). A time-based correction of this prediction using the nearest neighbors is also included in this second step.

**Drawbacks:**

* It can’t discuss to know how our model can evolve in the case where unknown attack occurs with all types of attacks by popular machine learning algorithms.
* It can’t describe each categorized of DOS attacks like back, Neptune etc. based on the network connections.
* Algorithm prediction results by best accuracy of classification algorithms with classification report of precision, recall and f1-score and additionally, to categorized other attacks of network connections.

**PROPOSED SYSTEM**

## Exploratory Data Analysis

This analysis is not meant to be providing a final conclusion on the reasons leading to network sector as it doesn't involve using any inferential statistics techniques/machine learning algorithms. Machine learning supervised classification algorithms will be used to give the network connection dataset and extract patterns, which would help in predicting the likely patient affected or not, thereby helping the attack of avoids for making better decisions in the future. Multiple datasets from different sources would be combined to form a generalized dataset, and then different machine learning algorithms would be applied to extract patterns and to obtain results with maximum accuracy.

## Data Wrangling

## In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

**Data collection**

The data set collected for predicting the network attacks is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, logistic, Decision tree algorithms, K-Nearest Neighbor (KNN) and Support vector classifier (SVC) are applied on the Training set and based on the test result accuracy, Test set prediction is done.

**Preprocessing**

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done. The correlation among attributes can be identified using plot diagram in data visualization process. Data preprocessing is the most time consuming phase of a data mining process. Data cleaning of connections, data removed several attributes that has no significance about the behavior of a packet transfers. Data integration, data reduction and data transformation are also to be applicable for network connections dataset. For easy analysis, the data is reduced to some minimum amount of records. Initially the Attributes which are critical to make a loan credibility prediction is identified with information gain as the attribute-evaluator and Ranker as the search-method.

**Environmental Requirements:**

1. Software Requirements:

Operating System : Windows

Tool : Anaconda with Jupyter Notebook

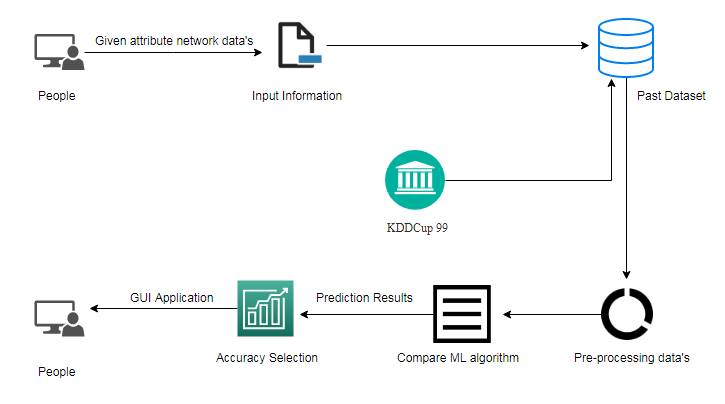
2. Hardware requirements:

Processor : Pentium IV/III

Hard disk : minimum 80 GB

RAM : minimum 2 GB

**Design architecture:**

****

**Future Work**

* Network sector want to automate the detecting the attacks of packet transfers from eligibility process (real time) based on the connection detail.
* To automate this process by show the prediction result in web application or desktop application.
* To optimize the work to implement in Artificial Intelligence environment.

**FEASIBILITY STUDY**

## Introduction

A feasibility analysis involves a detailed assessment of the need, value and practicality of a p systems development... Feasibility analysis n forms the transparent decisions at crucial points during the developmental process as we determine whether it is operationally, economically and technically realistic to proceed with a particular course of action.

Feasibility analysis can be used in each of the steps to assess the financial, technical and operational capacity to proceed with particular activities.

### Types of feasibility

A feasibility analysis usually involves a thorough assessment of the financial (value), technical (practicality), and operational (need) aspects of a proposal. In systems development projects, business managers are primarily responsible for assessing the operational feasibility of the system, and information technology (IT) analysts are responsible for assessing technical feasibility. Both then work together to prepare a cost–benefit analysis of the proposed system to determine its economic feasibility.

#### Operational feasibility

A systems development project is likely to be operationally feasible if it meets the 'needs' and expectations of the organisation. User acceptance is an important determinant of operational feasibility. It requires careful consideration of:

corporate culture;

staff resistance or receptivity to change;

management support for the new system;

the nature and level of user involvement in the development and implementation of the system; direct and indirect impacts of the new system on work practices;

anticipated performance and outcomes of the new system compared with the existing system;

training requirements and other change management strategies; and

‘pay back’ periods (ie trade-off between long-term organisational benefits and short-term inefficiencies during system development and implementation).

#### Technical feasibility

A systems development project may be regarded as technically feasible or practical if the organization has the necessary expertise and infrastructure to develop, install, operate and maintain the proposed system. Organizations will need to make this assessment based on:

Knowledge of current and emerging technological solutions;

Availability of technically qualified staff in-house for the duration of the project and subsequent maintenance phase;

Availability of infrastructure in-house to support the development and maintenance of the proposed system;

Where necessary, the financial and/or technical capacity to procure appropriate infrastructure and expertise from outside;

Capacity of the proposed system to accommodate increasing levels of use over the medium term;

The capacity of the proposed system to meet initial performance expectations and accommodate new functionality over the medium term.

#### Economic feasibility

A systems development project may be regarded as economically feasible or good value to the organization if its anticipated benefits outweigh its estimated costs. Many development costs are easier to identify. These costs may include the time, budget and staff resources invested during the design and implementation phase, as well as infrastructure, support, training and maintenance costs incurred after implementation. Nonetheless, it can also be difficult to accurately quantify project costs when new technologies and complex systems are involved. In these high-risk situations it may be appropriate to use sophisticated cost-benefit analysis tools to make appropriate assessments of financial feasibility.

**SOFTWARE ENGINEERING PARADIGM APPLIED:**

As per the Software development lifecycle,

1. The requirement specification of the project is collected properly.

2. Feasibility studies are taken upon the requirement collected to test the feasibility of the project.

3. According to the requirement the project gets modularized.

4. Architectural and detailed designs are drawn for each and every module.

5. During implementation phase, comments are used to describe the actions that are taking place, which is important for documentation and future reference.

6. Testing is extensively done to check whether the project is doing the task for what it has been designed.

* 1. **Security Feasibility**

It is that a third person watching the communication between the sender and the receiver will not be able to find out whether the sender has been active, and when, in the sense that he really embedded a message in the cover-text. In other words, stegotexts should be indistinguishable from covertexts.

* 1. **System Feasibility**

The proposed system can be developed using the present hardware and software technologies. The project requires following requirements.

Hardware and software architecture with minimum requirements, which supports an operating system on which Java toolkit and Media player applications can be developed and deployed. The estimated time given to different phases in the project such as Analysis, Design, implementation and testing all sum up to make a total time required to complete the project as approximately equal to 2 months (Excluding future Enhancement).

* 1. **Special Features**

For video, a combination of sound and image techniques can be used. This is due to the fact that video generally has separate inner files for the video (consisting of many images) and the sound. So techniques can be applied in both areas to hide data. Due to the size of video files, the scope for adding lots of data is much greater and therefore the chances of hidden data being detected is quite low.

Bottom of Form

Bottom of Form

**Gantt chart**

45

25

20

50

Days-180

40

Analysis

Design

Implementation

Testing

Maintenance

Completed

Total Number of days to complete: **180 Days**

**Implementation**

Create/ Delete User

Admin Module

**Testing**

20

25

**Design**

All Type of **Maintenance**

Unit Testing /

I/O Testing

50

45

Transaction Details and

All operation

Login/Logout

40

**Analysis**

**LOC – BASED ESTIMATION**

**AREA OF CODE OPSTEMISTIC MOST LIKELY PESSIMISTIC LOC-EV**

Client Programming 2200 3250 3700 3200

Network Programming 1250 1500 1450 1350

4550

The three points or expected value is computed as

EV = (S opt + 4 Sm + S Press ) / 6

Thus the estimated lines of code(LOC) are 4550

Assuming the organization average productivity for system of this type is 1620 LOC/pm

Based on the burden labor rate of Rs. 7000 per month, the cost per line of code is approximately Rs. 4.00

Based on the LOC estimate,

The estimation effort e is computed as

E = estimated LOC/(LOC / pm)

= 4550 / 1620 = 2.8086

app – 4 persons in a month or 1 person in 4 month

**Software Description:**

Anaconda is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) distribution of the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [R](https://en.wikipedia.org/wiki/R_(programming_language)) programming languages for [scientific computing](https://en.wikipedia.org/wiki/Scientific_computing) ([data science](https://en.wikipedia.org/wiki/Data_science), [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications, large-scale data processing, [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics), etc.), that aims to simplify [package management](https://en.wikipedia.org/wiki/Package_management) and deployment. Package versions are managed by the [package management system](https://en.wikipedia.org/wiki/Package_manager) “Conda”. The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS. So, Anaconda distribution comes with more than 1,400 packages as well as the [Conda](https://en.wikipedia.org/wiki/Conda_(package_manager)) package and virtual environment manager called Anaconda Navigator and it eliminates the need to learn to install each library independently. The open source packages can be individually installed from the Anaconda repository with the conda install command or using the pip install command that is installed with Anaconda. [Pip packages](https://en.wikipedia.org/wiki/Pip_(package_manager)) provide many of the features of conda packages and in most cases they can work together. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index) or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda.

Anaconda Navigator:

Anaconda Navigator is a desktop [graphical user interface (GUI)](https://en.wikipedia.org/wiki/Graphical_user_interface) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using [command-line commands](https://en.wikipedia.org/wiki/Command-line_interface). Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux).

The following applications are available by default in Navigator:

* [JupyterLab](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Lab)
* [Jupyter Notebook](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Notebook)
* [QtConsole](https://qtconsole.readthedocs.io/en/latest/)
* [Spyder](https://en.wikipedia.org/wiki/Spyder_(software))
* [Glueviz](http://glueviz.org/)
* [Orange](https://en.wikipedia.org/wiki/Orange_(software))
* [Rstudio](https://en.wikipedia.org/wiki/Rstudio)
* [Visual Studio Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)

### Conda:

Conda is an [open source](https://en.wikipedia.org/wiki/Open-source_software), [cross-platform](https://en.wikipedia.org/wiki/Cross-platform), language-agnostic [package manager](https://en.wikipedia.org/wiki/Package_manager) and environment management system that installs, runs and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., [R](https://en.wikipedia.org/wiki/R_(programming_language))), including multi-languages. The Conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

### The Jupyter Notebook:

#### The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

## [Notebook document](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#id5):

Notebook documents (or “notebooks”, all lower case) are documents produced by the [Jupyter Notebook App](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-app), which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc…). Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc.) as well as executable documents which can be run to perform data analysis.

## [Jupyter Notebook App](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#id6):

The Jupyter Notebook App is a server-client application that allows editing and running [notebook documents](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-document) via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet. In addition to displaying/editing/running notebook documents, the Jupyter Notebook App has a “Dashboard” ([Notebook Dashboard](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#dashboard)), a “control panel” showing local files and allowing to open notebook documents or shutting down their [kernels](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#kernel).

## [kernel](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#id7):

A notebook kernel is a “computational engine” that executes the code contained in a [Notebook document](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-document). The ipython kernel, referenced in this guide, executes python code. Kernels for many other languages exist ([official kernels](http://jupyter.readthedocs.org/en/latest/#kernels)). When you open a [Notebook document](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-document), the associated kernel is automatically launched. When the notebook is executed (either cell-by-cell or with menu Cell -> Run All), the kernel performs the computation and produces the results. Depending on the type of computations, the kernel may consume significant CPU and RAM. Note that the RAM is not released until the kernel is shut-down.

## [Notebook Dashboard](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#id8):

The Notebook Dashboard is the component which is shown first when you launch [Jupyter Notebook App](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-app). The Notebook Dashboard is mainly used to open [notebook documents](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-document), and to manage the running [kernels](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#kernel) (visualize and shutdown). The Notebook Dashboard has other features similar to a file manager, namely navigating folders and renaming/deleting files.

Working Process:

* Download and install anaconda and get the most useful package for machine learning in Python.
* Load a dataset and understand its structure using statistical summaries and data visualization.
* machine learning models, pick the best and build confidence that the accuracy is reliable.

Python is a popular and powerful interpreted language. Unlike R, Python is a complete language and platform that you can use for both research and development and developing production systems. There are also a lot of modules and libraries to choose from, providing multiple ways to do each task. It can feel overwhelming.

The best way to get started using Python for machine learning is to complete a project.

* It will force you to install and start the Python interpreter (at the very least).
* It will give you a bird’s eye view of how to step through a small project.
* It will give you confidence, maybe to go on to your own small projects.

When you are applying machine learning to your own datasets, you are working on a project. A machine learning project may not be linear, but it has a number of well-known steps:

* Define Problem.
* Prepare Data.
* Evaluate Algorithms.
* Improve Results.
* Present Results.

The best way to really come to terms with a new platform or tool is to work through a machine learning project end-to-end and cover the key steps. Namely, from loading data, summarizing data, evaluating algorithms and making some predictions.

Here is an overview of what we are going to cover:

1. Installing the Python anaconda platform.
2. Loading the dataset.
3. Summarizing the dataset.
4. Visualizing the dataset.
5. Evaluating some algorithms.
6. Making some predictions.

### Model Selection:

This is the most exciting phase in Applying Machine Learning to any Dataset. It is also known as Algorithm selection for Predicting the best results. Usually Data Scientists use different kinds of Machine Learning algorithms to the large data sets. But, at high level all those different algorithms can be classified in two groups : supervised learning and unsupervised learning. Supervised learning : Supervised learning is a type of system in which both input and desired output data are provided. Input and output data are labeled for classification to provide a learning basis for future data processing. Supervised learning problems can be further grouped into **Regression** and **Classification** problems.

A **regression** problem is when the output variable is a real or continuous value, such as “salary” or “weight”. A **classification** problem is when the output variable is a category like filtering emails “spam” or “not spam”

Unsupervised Learning : Unsupervised learning is the [algorithm](https://whatis.techtarget.com/definition/algorithm) using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. In our dataset we have the outcome variable or Dependent variable i.e Y having only two set of values, either M (Malign) or B(Benign). So we will use Classification algorithm of supervised learning.

**Modules:**

* Data validation and pre-processing technique (Module-01)
* Exploration data analysis of visualization and training a model by given attributes (Module-02)
* Performance measurements of logistic regression and decision tree algorithms (Module-03)
* Performance measurements of Support vector classifier and Random forest (Module-04)
* Performance measurements of KNN and Naive Bayes (Module-05)
* GUI based prediction of crop yield and yield cost (Module-06)

**Module-01/02:**

Variable Identification Process / data validation process:

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers uses this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model. For example, time series data can be analyzed by regression algorithms; classification algorithms can be used to analyze discrete data. (For example to show the data type format of given dataset)



Fig: Given data frame

Data Validation/ Cleaning/Preparing Process:

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.



Fig: Spliting the given dataset

Data Pre-processing:

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format; for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set.

**Module-03:**

Exploration data analysis of visualization:

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

* How to chart time series data with line plots and categorical quantities with bar charts.
* How to summarize data distributions with histograms and box plots.
* How to summarize the relationship between variables with scatter plots.

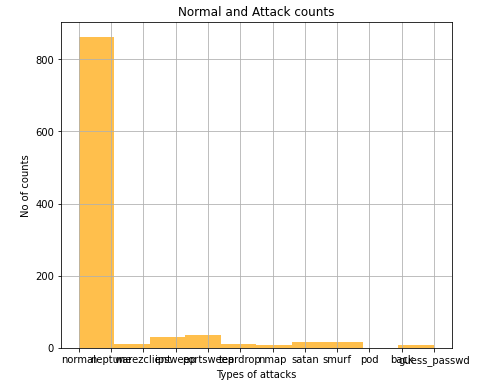


Fig: different types of attack counts

Many machine learning algorithms are sensitive to the range and distribution of attribute values in the input data. Outliers in input data can skew and mislead the training process of machine learning algorithms resulting in longer training times, less accurate models and ultimately poorer results.

Even before predictive models are prepared on training data, outliers can result in misleading representations and in turn misleading interpretations of collected data. Outliers can skew the summary distribution of attribute values in descriptive statistics like mean and standard deviation and in plots such as histograms and scatterplots, compressing the body of the data. Finally, outliers can represent examples of data instances that are relevant to the problem such as anomalies in the case of fraud detection and computer security.

It couldn’t fit the model on the training data and can’t say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from the data, and it is not getting up too much noise. Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

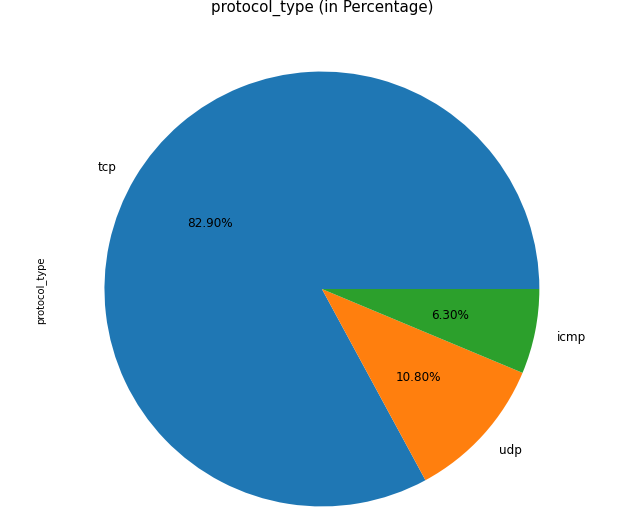
****

Fig: Percentage level of crop yield production by state

**Module-04:**

[**Logistic Regression**](https://en.wikipedia.org/wiki/Logistic_regression)**:**

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts P(Y=1) as a function of X. Logistic regression Assumptions:

* Binary logistic regression requires the dependent variable to be binary.
* For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
* Only the meaningful variables should be included.
* The independent variables should be independent of each other. That is, the model should have little.
* The independent variables are linearly related to the log odds.
* Logistic regression requires quite large sample sizes.

**Module-05:**

[Decision Tree](https://www.geeksforgeeks.org/decision-tree/):

It is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. Assumptions of Decision tree:

* At the beginning, we consider the whole training set as the root.
* Attributes are assumed to be categorical for information gain, attributes are assumed to be continuous.
* On the basis of attribute values records are distributed recursively.
* We use statistical methods for ordering attributes as root or internal node.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed.

This process is continued on the training set until meeting a termination condition. It is constructed in a top-down recursive divide-and-conquer manner. All the attributes should be categorical. Otherwise, they should be discretized in advance. Attributes in the top of the tree have more impact towards in the classification and they are identified using the information gain concept. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers.

**Module-06:**

**Support Vector Machines (SVM):**

A classifier that categorizes the data set by setting an optimal hyper plane between data. I chose this classifier as it is incredibly versatile in the number of different kernelling functions that can be applied and this model can yield a high predictability rate. Support Vector Machines are perhaps one of the most popular and talked about machine learning algorithms. They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high-performing algorithm with little tuning.

* How to disentangle the many names used to refer to support vector machines.
* The representation used by SVM when the model is actually stored on disk.
* How a learned SVM model representation can be used to make predictions for new data.
* How to learn an SVM model from training data.
* How to best prepare your data for the SVM algorithm.
* Where you might look to get more information on SVM.

**Random Forest:**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees’ habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning). Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The [random forest](https://en.wikipedia.org/wiki/Random_forest) algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

* Pick N random records from the dataset.
* Build a decision tree based on these N records.
* Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
* In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

**Module-07:**

Tkinter is a python library for developing GUI (Graphical User Interfaces). We use the tkinter library for creating an application of UI (User Interface), to create windows and all other graphical user interface and Tkinter will come with Python as a standard package, it can be used for security purpose of each users or accountants. There will be two kinds of pages like registration user purpose and login entry purpose of users.

Parameter calculations:

Accuracy calculation:

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

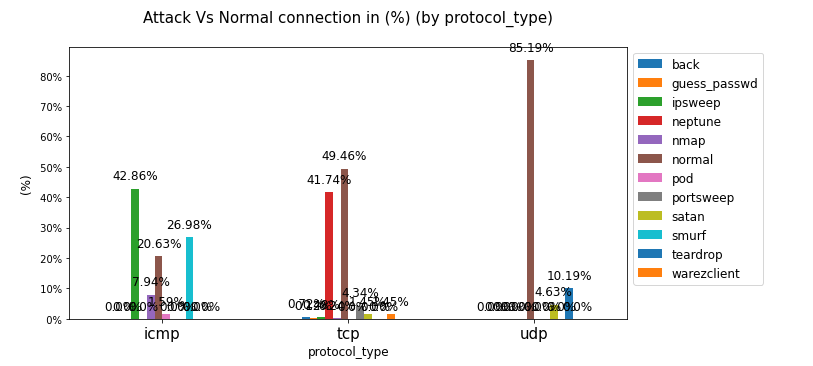
False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

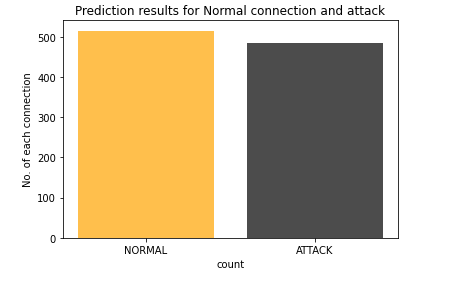
True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

It achieved precision, recall, true positive rate (TPR), and false positive rate (FPR) for each classification techniques as it is shown in the above tables and also achieved different interesting confusion matrix for each classification techniques and we can see the classification performance of each classifiers by the help of confusion matrix. We use a confusion matrix to compute the accuracy rate of each severity class. For each class, it demonstrates how instances from that class receive the various classifications. Here in the next table we have shown instances that are correctly classified and incorrectly classified in accordance with overall accuracy of each classification techniques. All classifiers perform similarly well with respect to the number of correctly classified instances.

Comparing Algorithm with prediction in the form of best accuracy result:

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies. 

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.



In the example below 4 different algorithms are compared:

* Logistic Regression
* Decision tree
* Support Vector Machines
* Now, the dimensions of new features in a numpy array called ‘n’ and it want to predict the species of this features and to do using the predict method which takes this array as input and spits out predicted target value as output.
* So, the predicted target value comes out to be 0. Finally to find the test score which is the ratio of no. of predictions found correct and total predictions made and finding accuracy score method which basically compares the actual values of the test set with the predicted values.

Sensitivity:

Sensitivity is a measure of the proportion of actual positive cases that got predicted as positive (or true positive). Sensitivity is also termed as Recall. This implies that there will be another proportion of actual positive cases, which would get predicted incorrectly as negative (and, thus, could also be termed as the false negative). This can also be represented in the form of a false negative rate. The sum of sensitivity and false negative rate would be 1. Let's try and understand this with the model used for predicting whether a person is suffering from the disease. Sensitivity is a measure of the proportion of people suffering from the disease who got predicted correctly as the ones suffering from the disease. In other words, the person who is unhealthy actually got predicted as unhealthy.

Mathematically, sensitivity can be calculated as the following:

Sensitivity = (True Positive) / (True Positive + False Negative)

The following is the details in relation to True Positive and False Negative used in the above equation.

* True Positive = Persons predicted as suffering from the disease (or unhealthy) are actually suffering from the disease (unhealthy); In other words, the true positive represents the number of persons who are unhealthy and are predicted as unhealthy.
* False Negative = Persons who are actually suffering from the disease (or unhealthy) are actually predicted to be not suffering from the disease (healthy). In other words, the false negative represents the number of persons who are unhealthy and got predicted as healthy. Ideally, we would seek the model to have low false negatives as it might prove to be life-threatening or business threatening.

The higher value of sensitivity would mean higher value of true positive and lower value of false negative. The lower value of sensitivity would mean lower value of true positive and higher value of false negative. For healthcare and financial domain, models with high sensitivity will be desired.

Specificity:

Specificity is defined as the proportion of actual negatives, which got predicted as the negative (or true negative). This implies that there will be another proportion of actual negative, which got predicted as positive and could be termed as false positives. This proportion could also be called a false positive rate. The sum of specificity and false positive rate would always be 1. Let's try and understand this with the model used for predicting whether a person is suffering from the disease. Specificity is a measure of the proportion of people not suffering from the disease who got predicted correctly as the ones who are not suffering from the disease. In other words, the person who is healthy actually got predicted as healthy is specificity.

Mathematically, specificity can be calculated as the following:

Specificity = (True Negative) / (True Negative + False Positive)

The following is the details in relation to True Negative and False Positive used in the above equation.

* True Negative = Persons predicted as not suffering from the disease (or healthy) are actually found to be not suffering from the disease (healthy); In other words, the true negative represents the number of persons who are healthy and are predicted as healthy.
* False Positive = Persons predicted as suffering from the disease (or unhealthy) are actually found to be not suffering from the disease (healthy). In other words, the false positive represents the number of persons who are healthy and got predicted as unhealthy.

The higher value of specificity would mean higher value of true negative and lower false positive rate. The lower value of specificity would mean lower value of true negative and higher value of false positive.

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. We need the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

True Positive Rate(TPR) = TP / (TP + FN)

False Positive rate(FPR) = FP / (FP + TN)

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

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

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct. (When the model predicts default: how often is correct?)

Precision = TP / (TP + FP)

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = TP / (TP + FN)

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Scoreis the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

General Formula:

F- Measure = 2TP / (2TP + FP + FN)

F1-Score Formula:

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

**Used Python Packages:**

**sklearn:**

* + In python, sklearn is a machine learning package which include a lot of ML algorithms.
  + Here, we are using some of its modules like train\_test\_split, DecisionTreeClassifier or Logistic Regression and accuracy\_score.

**NumPy:**

* + It is a numeric python module which provides fast maths functions for calculations.
  + It is used to read data in numpy arrays and for manipulation purpose.

**Pandas:**

* + Used to read and write different files.
  + Data manipulation can be done easily with data frames.

**Matplotlib:**

* + Data visualization is a useful way to help with identify the patterns from given dataset.
  + Data manipulation can be done easily with data frames.

**tkinter:**

* + Standard python interface to the GUI toolkit.
  + Accessible to everybody and reusable in various contexts.

**LITERATURE SURVEY**

1. **THE FEASIBILITY OF LAUNCHING AND DETECTING JAMMING ATTACKS IN WIRELESS NETWORKS**

**AUTHORS:** W. Xu, W. Trappe, Y. Zhang, and T. Wood

**ABSTRACT**

Wireless networks are built upon a shared medium that makes it easy for adversaries to launch jamming-style attacks. These attacks can be easily accomplished by an adversary emitting radio frequency signals that do not follow an underlying MAC protocol. Jamming attacks can severely interfere with the normal operation of wireless networks and, consequently, mechanisms are needed that can cope with jamming attacks. In this paper, we examine radio interference attacks from both sides of the issue: first, we study the problem of conducting radio interference attacks on wireless networks, and second we examine the critical issue of diagnosing the presence of jamming attacks. Specifically, we propose four different jamming attack models that can be used by an adversary to disable the operation of a wireless network, and evaluate their effectiveness in terms of how each method affects the ability of a wireless node to send and receive packets. We then discuss different measurements that serve as the basis for detecting a jamming attack, and explore scenarios where each measurement by itself is not enough to reliably classify the presence of a jamming attack. In particular, we observe that signal strength and carrier sensing time are unable to conclusively detect the presence of a jammer. Further, we observe that although by using packet delivery ratio we may differentiate between congested and jammed scenarios, we are nonetheless unable to conclude whether poor link utility is due to jamming or the mobility of nodes. The fact that no single measurement is sufficient for reliably classifying the presence of a jammer is an important observation, and necessitates the development of enhanced detection schemes that can remove ambiguity when detecting a jammer. To address this need, we propose two enhanced detection protocols that employ consistency checking. The first scheme employs signal strength measurements as a reactive consistency check for poor packet delivery ratios, while the second scheme employs location information to serve as the consistency check. Throughout our discussions, we examine the feasibility and effectiveness of jamming attacks and detection schemes using the MICA2 Mote platform.

1. **AN INTEGRATED APPROACH FOR JAMMER DETECTION USING SOFTWARE DEFINED RADIO**

**AUTHORS**: R. Bhojani and R. Joshi

**ABSTRACT**

Due to shared nature of wireless communication any malicious user can easily monitored communication between two devices and emits false message to block communication. Nowadays increased use of software defined radio (SDR) technology makes any types of jammer device using same hardware with little modification in software. A jammer transmits radio signal to block legitimate communication either overlapping signal with more power or reducing signal to noise ratio. In this paper we have survey different jammer detection methods for efficient detection of jammers presence in system. Existing jammer detection methods like packet delivery ratio (PDR), packet send ratio (PSR), bad packet ratio (BPR) and signal to noise ratio (SNR) can effectively detects jammer, here we have proposed novel method for jammer detection using communication parameter used in SDR like synchronization indicator, iteration and adaptive signal to jammer plus noise ratio (ASNJR). This system uses that parameter which is readily available in system so computation has been reduced and ASNJR also has been adaptively updated with and without presence of jammer. Experimental result show that this system based on SDR effectively detects presence of jammer.

1. **AN SVM-BASED METHODFOR CLASSIFICATION OF EXTERNAL INTERFERENCE IN INDUSTRIAL WIRELESS SENSOR AND ACTUATOR NETWORKS**

**AUTHORS:** S. Grimaldi, A. Mahmood, and M. Gidlund

**ABSTRACT**

In recent years, the adoption of industrial wireless sensor and actuator networks (IWSANs) has greatly increased. However, the time-critical performance of IWSANs is considerably affected by external sources of interference. In particular, when an IEEE 802.11 network is coexisting in the same environment, a significant drop in communication reliability is observed. This, in turn, represents one of the main challenges for a wide-scale adoption of IWSAN. Interference classification through spectrum sensing is a possible step towards interference mitigation, but the long sampling window required by many of the approaches in the literature undermines their run-time applicability in time-slotted channel hopping (TSCH)-based IWSAN. Aiming at minimizing both the sensing time and the memory footprint of the collected samples, a centralized interference classifier based on support vector machines (SVMs) is introduced in this article. The proposed mechanism, tested with sample traces collected in industrial scenarios, enables the classification of interference from IEEE 802.11 networks and microwave ovens, while ensuring high classification accuracy with a sensing duration below 300 ms. In addition, the obtained results show that the fast classification together with a contained sampling frequency ensure the suitability of the method for TSCH-based IWSAN.

1. **A TRAINING ALGORITHM FOR OPTIMAL MARGIN CLASSIFIERS**

**AUTHORS:** B. E. Boser, I. M. Guyon, and V. N. Vapnik

**ABSTRACT**

A training algorithm that maximizes the margin between the training patterns and the decision boundary is presented. The technique is applicable to a wide variety of the classification functions, including Perceptrons, polynomials, and Radial Basis Functions. The effective number of parameters is adjusted automatically to match the complexity of the problem. The solution is expressed as a linear combination of supporting patterns. These are the subset of training patterns that are closest to the decision boundary. Bounds on the generalization performance based on the leave-one-out method and the VC-dimension are given. Experimental results on optical character recognition problems demonstrate the good generalization obtained when compared with other learning algorithms.

1. **ONE-AGAINST-ALL MULTI-CLASS SVM CLASSIFICATION USING RELIABILITY MEASURES**

**AUTHORS:** Y. Liu and Y. F. Zheng

**ABSTRACT**

Support vector machines (SVM) is originally designed for binary classification. To extend it to multi-class scenario, a typical conventional way is to decompose an M-class problem into a series of two-class problems, for which one-against-all is the earliest and one of the most widely used implementations. However, certain theoretical analysis reveals a drawback, i.e., the competence of each classifier is totally neglected when the results of classification from the multiple classifiers are combined for the final decision. To overcome this limitation, this paper introduces reliability measures into the multi-class framework. Two measures are designed: static reliability measure (SRM) and dynamic reliability measure (DRM). SRM works on a collective basis and yields a constant value regardless of the location of the test sample. DRM, on the other hand, accounts for the spatial variation of the classifier's performance. Based on these two reliability measures, a new decision strategy for the one-against-all method is proposed, which is tested on benchmark data sets and demonstrates its effectiveness.

**DESIGN**

**ARCHITECTURE**

KDD CUP data

Test dataset

Data Processing

Training dataset

Classification ML Algorithm

Model

**FLOW DIAGRAM**

**Construction of a Predictive Model**

## Machine learning needs data gathering have lot of past data’s. Data gathering have sufficient historical data and raw data. Before data pre-processing, raw data can’t be used directly. It’s used to preprocess then, what kind of algorithm with model. Training and testing this model working and predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy.

Data Gathering

Data Pre-Processing

Choose model

Train model

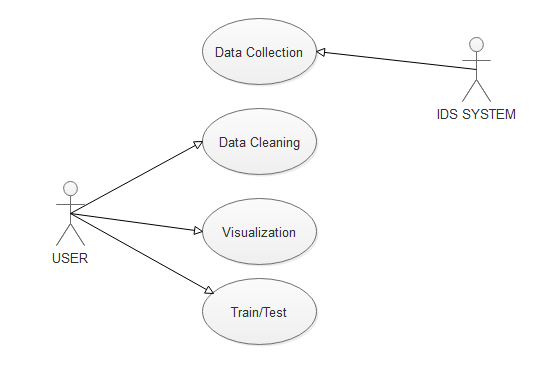
Test model

Tune model

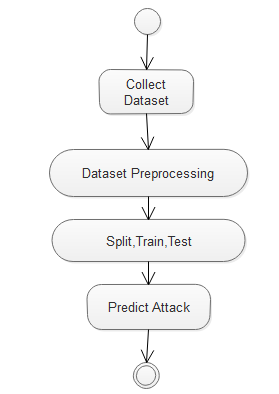
Prediction

Fig: Process of dataflow diagram

**USE CASE DIAGRAM**

****

**ACTIVITY DIAGRAM**

****

**SAMPLE CODE**

**PREPROCESSING**

import pandas as pd

train=pd.read\_csv('dataset/Train.txt',sep=',')

train.head()

train1=train.head(1000)

columns=["duration","protocol\_type","service","flag","src\_bytes","dst\_bytes","land",

"wrong\_fragment","urgent","hot","num\_failed\_logins","logged\_in",

"num\_compromised","root\_shell","su\_attempted","num\_root","num\_file\_creations",

"num\_shells","num\_access\_files","num\_outbound\_cmds","is\_host\_login",

"is\_guest\_login","count","srv\_count","serror\_rate", "srv\_serror\_rate",

"rerror\_rate","srv\_rerror\_rate","same\_srv\_rate", "diff\_srv\_rate","srv\_diff\_host\_rate","dst\_host\_count","dst\_host\_srv\_count","dst\_host\_same\_srv\_rate",

"dst\_host\_diff\_srv\_rate","dst\_host\_same\_src\_port\_rate",

"dst\_host\_srv\_diff\_host\_rate","dst\_host\_serror\_rate","dst\_host\_srv\_serror\_rate",

"dst\_host\_rerror\_rate","dst\_host\_srv\_rerror\_rate","attack", "last\_flag"]

train1.columns=columns

train1.head()

train1.to\_csv('dataset/train1.csv')

**LOGISTIC REGRESSION**

import pandas as p

import numpy as n

import matplotlib.pyplot as plt

import seaborn as s

from sklearn.linear\_model import LogisticRegression

#read the given dataset

df= p.read\_csv("dataset/finaltrain1.csv")

df.drop(columns=df.columns[:1], axis=1,inplace=True)

df.head()

x = df.iloc[ : , :-1].values

y = df.iloc[:, -1:].values

#spliting the dataset into training set and test set

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y,test\_size = 0.25, random\_state =0 )

logreg = LogisticRegression(random\_state=0,solver='lbfgs',multi\_class='multinomial')

logreg.fit( train\_X, train\_y)

result=logreg.predict(x\_test)

result

#confusion matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, result)

print(cm)

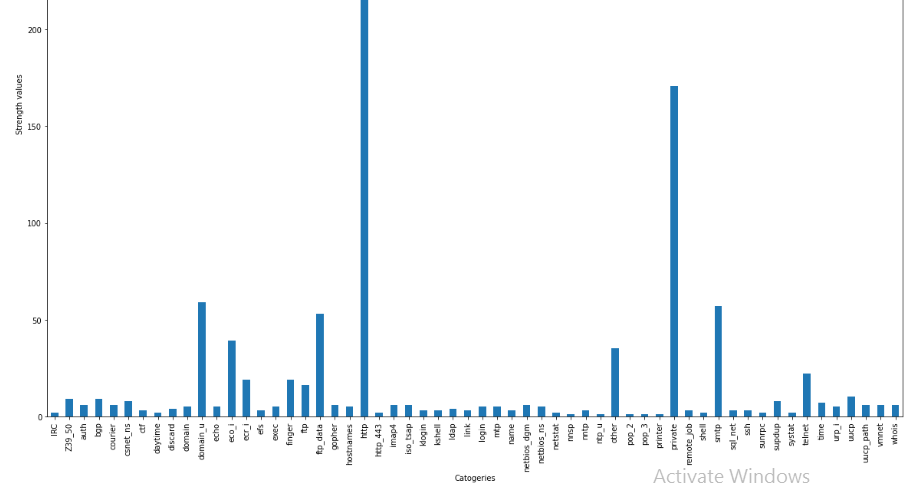
#pickle file joblib

import joblib

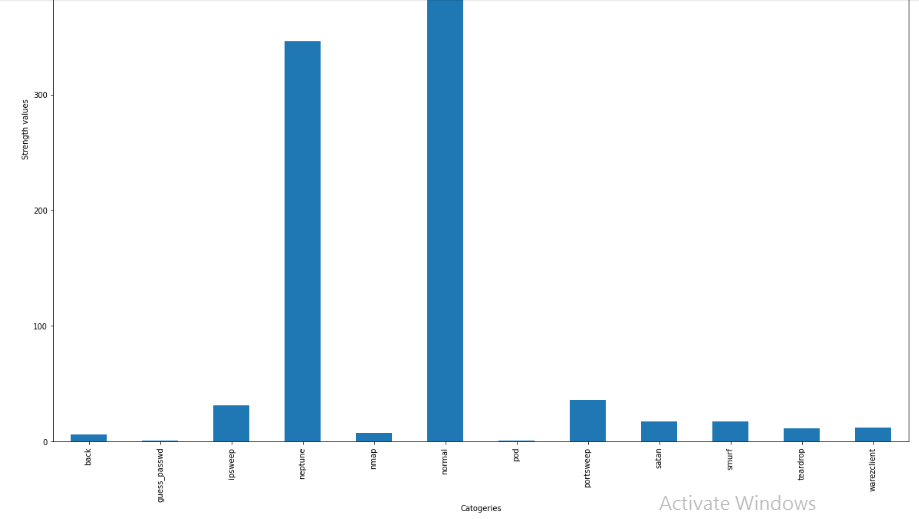
joblib.dump(logreg, 'final\_models/lr.pkl')

**SNAPSHOT**

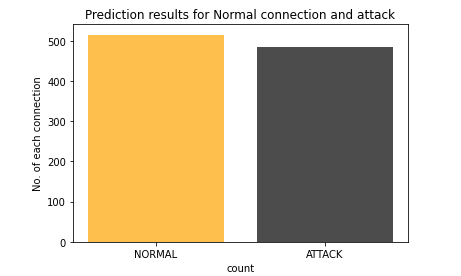
**Types of services**



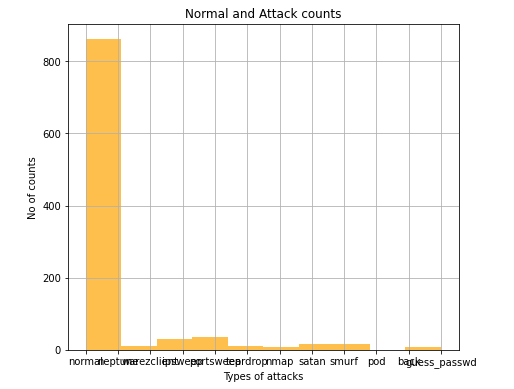
Types of attacks



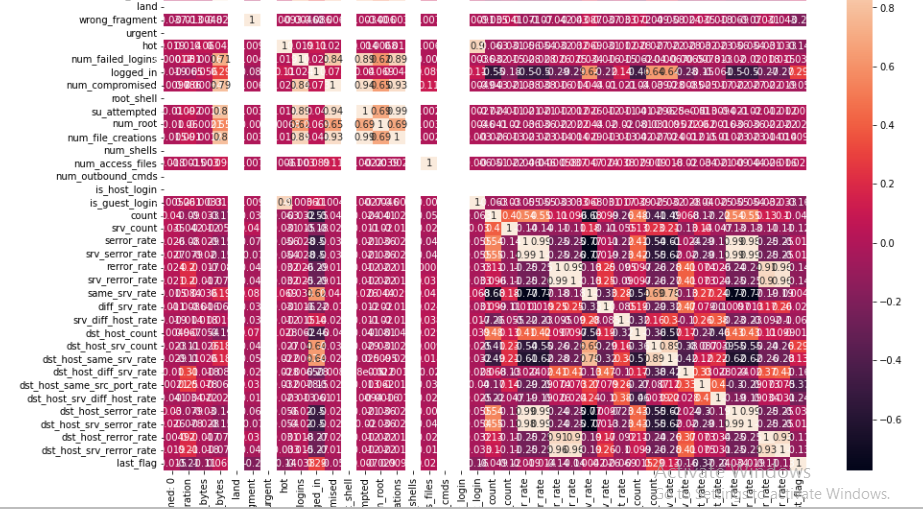
Normal VS Attack data

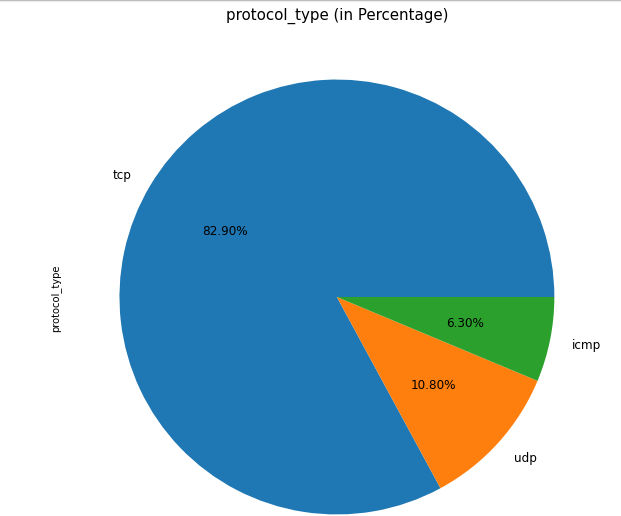


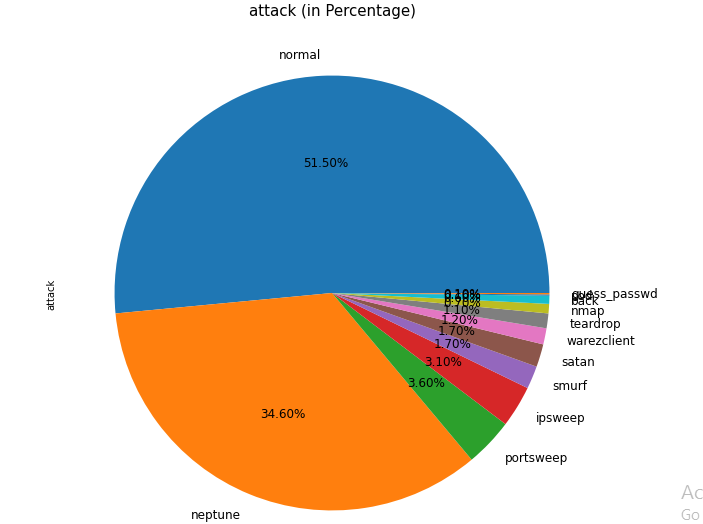
Histogram

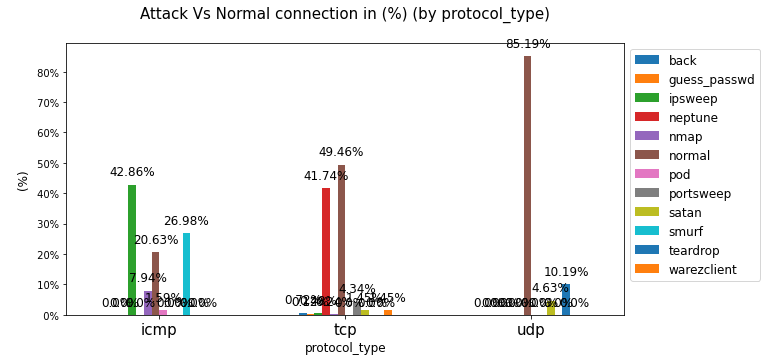


Heat Map









GUI based prediction

