**Intrusion Detection Systems**

**Introduction:**

Intrusion detection systems (IDSs) are software or hardware systems that automate the process of monitoring the events occurring in a computer system or network, analyzing them for signs of security problems. As network attacks have increased in number and severity over the past few years, intrusion detection systems have become a necessary addition to the security infrastructure of most organizations. This guidance document is intended as a primer in intrusion detection, developed for those who need to understand what security goals intrusion detection mechanisms serve, how to select and configure intrusion detection systems for their specific system and network environments, how to manage the output of intrusion detection systems, and how to integrate intrusion detection functions with the rest of the organizational security infrastructure.

**Modules:**

There will be 4 modules in this project:

1. **Scan log :** Browse the log file and submit Displays different types of attacks present in the log file and details.
2. **Web app scanner:** Enter web application url and submit Scanned result will display the vulnerability of that web application.
3. **Test query :** Enter query to test Displays attack type of query.
4. **Realtime IDS :**Insert web server log location and submit Realtime scan results are displayed. And also send an alert email that an attack is detected.

**Descriptions:**

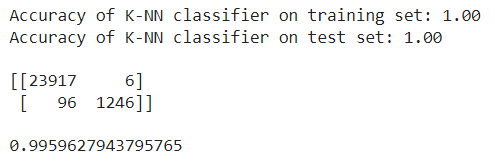
Machine Learning based real-time Intrusion detection system and Web Application security scanner, powered using Python-3 and flask. It wields defense against Cross-Site Scripting (XSS), Path Traversal attacks, Server Side Includes (SSI), OS Command Injections, XPath Injections, LDAP Injections, CRLF Injections, and other Anomalous categories.

**Tools & Technology:**

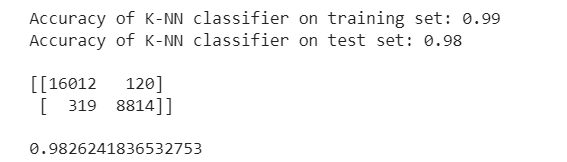
* Python-3
* Flask for Web Application (Backend)
* HTML, CSS, Bootstrap5, JavaScript – for implementing the frontend part (web development)
* SQL Database
* Machine learning for classification & pre processing of data sets which are taken from kaggle. Libraries are panda, numpy, and sklearn.Classifiers are K-Near Neighbors, Decision Tree, Random forest , Support Vector Machine, Logistic Regression, Gradient Boosting.
* Git for uploading projects on github.
* Google cloud for deployment.

**Attack Categories:**

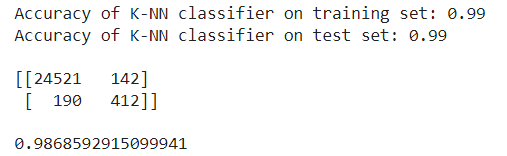
* **Cross-site Scripting(XSS):**  LR,SVM,GB,DT,KNN,RF from all this classifier KNN Accuracy is best of all so that is why KNN is selected and dumped into .sav model.



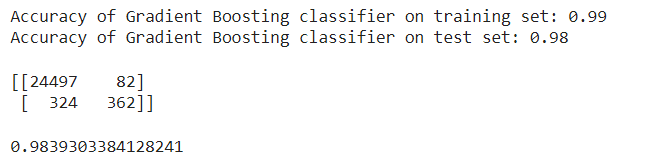
* **SQL injection:** LR,SVM,GB,DT,KNN,RF from all this classifier KNN Accuracy is best of all so that is why KNN is selected and dumped into .sav model.



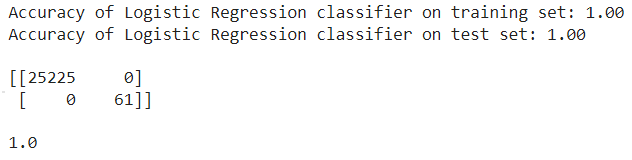
* **Path Traversal attacks:** LR,SVM,GB,DT,KNN,RF from all this classifier KNN Accuracy is best of all so that is why KNN is selected and dumped into .sav model.



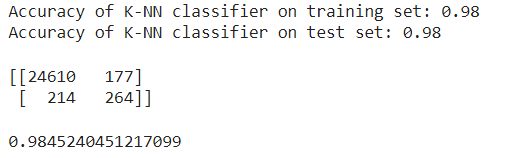
* **OS Command Injections:** LR,SVM,GB,DT,KNN,RF from all these classifiers KNN Accuracy is best of all, so that is why GB is selected and dumped into the .sav model.



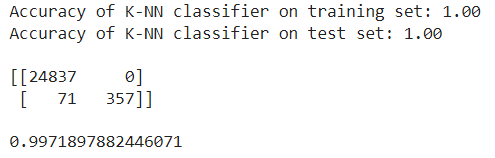
* **CRLF Injections:** LR,SVM,GB,DT,KNN,RF from all these classifiers KNN Accuracy is best of all so that is why LR is selected and dumped into the .sav model.



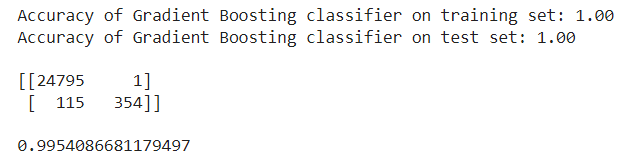
* **Server Side Includes (SSI):** LR,SVM,GB,DT,KNN,RF from all these classifiers KNN Accuracy is best of all so that is why KNN is selected and dumped into the .sav model.



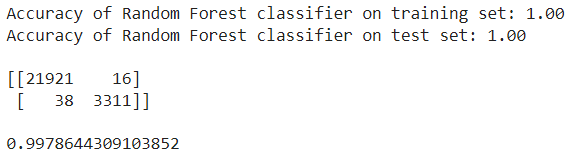
* **LDAP Injections:** LR,SVM,GB,DT,KNN,RF from all these classifiers KNN Accuracy is best of all so that is why KNN is selected and dumped into the .sav model.



* **XPath Injections** **:** LR,SVM,GB,DT,KNN,RF from all these classifiers KNN Accuracy is best of all so that is why GB is selected and dumped into the .sav model.



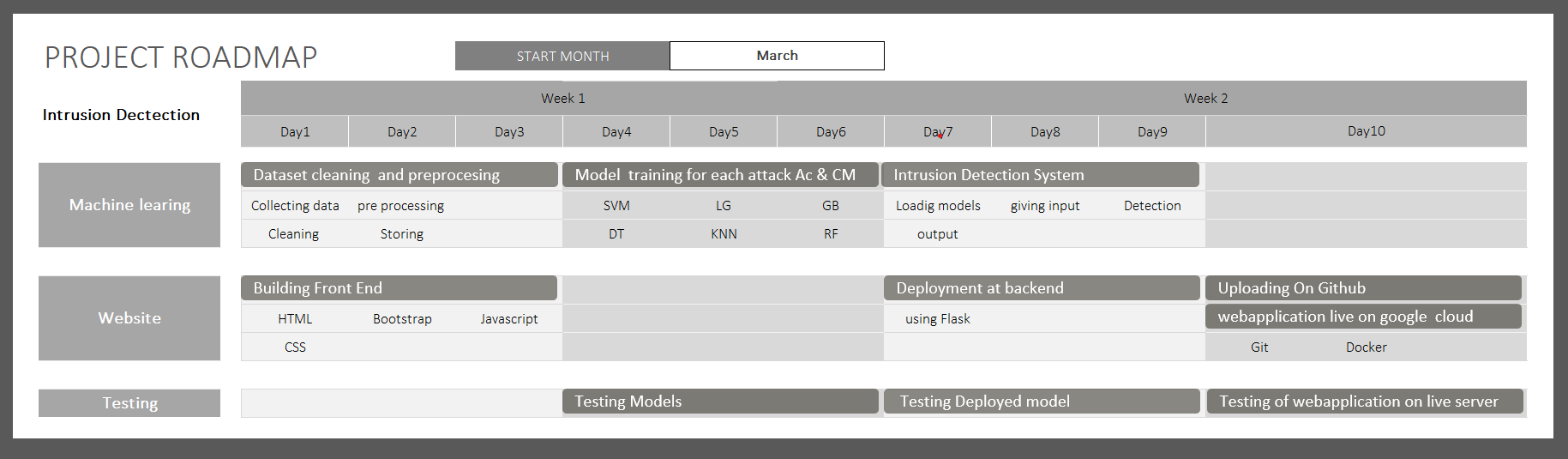
* **Anomalous** **:** LR,SVM,GB,DT,KNN,RF from all these classifiers KNN Accuracy is best of all so that is why RF is selected and dumped into the .sav model.



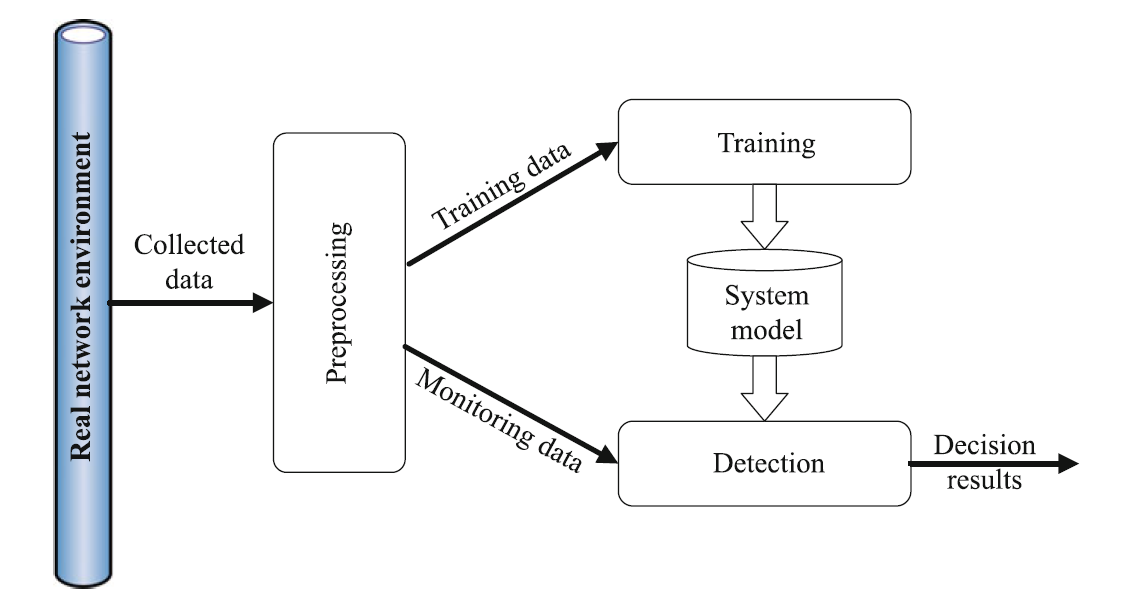
**Technology used from each stage:**

* Stage 1 -Object Oriented Programming & Data Structure
* Stage 2 - Database Management System & Structure Quarry Languege
* Stage 3 - Python3
* Stage 4 –HTML, CSS, Bootstrap
* Stage 5 –JavaScript
* Stage 7– Machine learning
* Stage 8– Flask
* Stage 9 – Docker/git

**Road Map:**



**Workflow:**



**Dataset:**

