PROJECT REPORT ON

ROBUST MALWARE DETECTION FOR INTERNET OF THINGS DEVICES USING DEEP EIGENSPACE LEARNING

CPT_S 534 Neutral Network Design & Appl

Robust Malware Detection for Internet Of (Battlefield) Things Devices Using Deep Eigenspace Learning
USER NAME
PASSWORD
Sign in

Submitted by
Rasumalla Khoushik Raj

ABSTRACT

Internet of Things (IoT) in military settings generally consists of a diverse range of Internet-connected devices and nodes (e.g. medical devices and wearable combat uniforms). These IoT devices and nodes are a valuable target for cyber criminals, particularly state-sponsored or nation state actors. A common attack vector is the use of malware. In this paper, we present a deep learning based method to detect Internet Of Battlefield Things (IoBT) malware via the device's Operational Code (OpCode) sequence. We transmute OpCodes into a vector space and apply a deep Eigenspace learning approach to classify malicious and benign applications. We also demonstrate the robustness of our proposed approach in malware detection and its sustainability against junk code insertion attacks. Lastly, we make available our malware sample on Github, which hopefully will benefit future research efforts (e.g. to facilitate evaluation of future malware detection approaches).

INTRODUCTION

Junk code injection attack is a malware anti-forensic technique against OpCode inspection. As the name suggests, junk code insertion may include addition of benign OpCode sequences, which do not run in a malware or inclusion of instructions (e.g. NOP) that do not actually make any difference in malware activities. Junk code insertion technique is generally designed to obfuscate malicious OpCode sequences and reduce the 'proportion' of malicious OpCodes in a malware In our proposed approach, we use an affinity based criteria to mitigate junk OpCode injection anti-forensics technique. Specifically, our feature selection method eliminates less instructive OpCodes to mitigate the effects of injecting junk OpCodes. To demonstrate the effectiveness of our proposed approach against code insertion attack, in an iterative manner, a specified proportion (f5%, 10%, 15%, 20%, 25%, 30%g) of all elements in each sample's generated graph were selected randomly and their value incremented by one. For example, in the 4th iteration of the evaluations, 20% of the indices in each sample's graph were chosen to increment their value by one. In addition, in our evaluations the possibility of a repetitive element selection was included to simulate injecting an OpCode more than once. Incrementing Ei;j in the sample's generated graph is equivalent to injecting OpCodej next to the OpCodei in a sample's instruction sequence to mislead the detection algorithm. Algorithm 2 describes an iteration of junk code insertion during experiments, and this procedure should repeat for each iteration of k-fold validation. To show the robustness of our proposed approach and benchmark it against existing proposals, two congruent algorithms described in Section 1 are applied on our generated dataset using Adaboost as the classification algorithm.

EXISTING SYSTEM:

Malware detection methods can be static or dynamic. In dynamic malware detection approaches, the program is executed in a controlled environment (e.g., a virtual machine or a sandbox) to collect its behavioral attributes such as required resources, execution path, and requested privilege, in order to classify a program as malware or benign. Static approaches (e.g., signature-based detection, byte-sequence n-gram analysis, opcode sequence identification and control flow graph traversal) statically inspect a program code to detect suspicious applications. David et al proposed Deepsign to automatically detect malware using a signature generation method. The latter creates a dataset based on behaviour logs of API calls, registry entries, web searches, port accesses, etc, in a sandbox and then converts logs to a binary vector. They used deep belief network for classification and reportedly achieved 98.6% accuracy. In another study, Pascanu et al. Proposed a method to model malware execution using natural language modeling. They extracted relevant features using recurrent neural network to predict the next API calls. Then, both logistic regression and multi-layer perceptrons were applied as the classification module on next API call prediction and using history of past events as features. It was reported that 98.3% true positive rate and 0.1% false positive rate were achieved. Demme et al. examined the feasibility of building a malware detector in IoT nodes' hardware using performance counters as a learning feature and K-Nearest Neighbor, Decision Tree and Random Forest as classifiers. The reported accuracy rate for different malware family ranges from 25% to 100%. Alam et al. applied Random Forest on a dataset of Internet-connected smartphone devices to recognize malicious codes. They executed APKs in an Android emulator and recorded different features such as memory information, permission and network for classification, and evaluated their approach using different tree sizes. Their findings showed that the optimal classifier contains 40 trees, and 0.0171 of mean square root was achieved.

PROPOSED SYSTEM:

Junk code injection attack is a malware anti-forensic technique against OpCode inspection. As the name suggests, junk code insertion may include addition of benign OpCode sequences, which do not run in a malware or inclusion of instructions (e.g. NOP) that do not actually make any difference in malware activities. Junk code insertion technique is generally designed to obfuscate malicious OpCode sequences and reduce the 'proportion' of malicious OpCodes in a malware In our proposed approach, we use an affinity based criteria to mitigate junk OpCode injection anti-forensics technique. Specifically, our feature selection method eliminates less instructive OpCodes to mitigate the effects of injecting junk OpCodes. To demonstrate the effectiveness of our proposed approach against code insertion attack, in an iterative manner, a specified proportion (f5%, 10%, 15%, 20%, 25%, 30%g) of all elements in each sample's generated graph were selected randomly and their value incremented by one. For example, in the 4th iteration of the evaluations, 20% of the indices in each sample's graph were chosen to increment their value by one. In addition, in our evaluations the possibility of a repetitive element selection was included to simulate injecting an OpCode more than once. Incrementing Ei;j in the sample's generated graph is equivalent to injecting OpCodej next to the OpCodei in a sample's instruction sequence to mislead the detection algorithm. Algorithm 2 describes an iteration of junk code insertion during experiments, and this procedure should repeat for each iteration of k-fold validation. To show the robustness of our proposed approach and benchmark it against existing proposals, two congruent algorithms described in Section 1 are applied on our generated dataset using Adaboost as the classification algorithm. All evaluations were conducted using MATLAB R2015a running on a Microsoft Windows 10 Pro personal computer powered by Intel Core i7 2.67GHz and 8GB RAM. A 10-fold cross validation was used in the validating, and the comparative summary is presented in Table 1. It is clear that our proposed approach outperforms the proposals of Hashemi et al.and Santos et al. . The approach of Santos et al. is a basic and commonly-known OpCode based malware detection algorithm and the approach of Hashemi et al. is the most similar in terms of using eigenspace as the basis. Accuracy is a general criteria for evaluating performance of an algorithm for both malware and benign class identification. The proposed approach achieves a high accuracy of 99.68%, while the approaches of Hashemi et al. and Santos et al.respectively achieve 98.59% and 95.91% accuracy. Recall or detection rate is an important criteria and the proposed approach achieves 98.37%, in comparison to 81.55% and 77.70% for the other two approaches. Our proposed approach also outperforms the approaches of Hashemi et al. and Santos et al., in terms of precision rate and F-Measure. Utilizing class-wise feature selection appears to result in beneficial features of minor class to be more effective during classification phase. Also, using Formulation to calculate OpCode's distance leads to the ability to represent more OpCode sequence patterns in the sample's graph. It also appears that employing deep neural networks for classification leads to a better classifier.

SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. **Three key considerations involved in the feasibility analysis are,**

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

MODULES

There are three modules can be divided here for this project they are listed as below

- User Activity
- Malware Deduction
- Junk Code Insertion Attacks

From the above three modules, project is implemented. Bag of discriminative words are achieved

MODULES:

1. **User Activity:**

User handling for some various times of IOT(internet of thinks example for Nest Smart Home, Kisi Smart Lock, Canary Smart Security System, DHL's IoT Tracking and Monitoring System,Cisco's Connected Factory,ProGlove's Smart Glove, Kohler Verdera Smart Mirror.If any kind of devices attacks for some unauthorized malware softwares.In this malware on threats for user personal dates includes for personal contact, bank account numbers and any kind of personal documents are hacking in possible.

2. **Malware Deduction**

Users search the any link notably, not all network traffic data generated by malicious apps correspond to malicious traffic. Many malware take the form of repackaged benign apps; thus, malware can also contain the basic functions of a benign app. Subsequently, the network traffic they generate can be characterized by mixed benign and malicious network traffic. We examine the traffic flow header using N-gram method from the natural language processing (NLP).

3. **Junk Code Insertion Attacks:**

Junk code injection attack is a malware anti-forensic technique against OpCode inspection. As the name suggests, junk code insertion may include addition of benign OpCode sequences, which do not run in a malware or inclusion of instructions (e.g. NOP) that do not actually make any difference in malware activities. Junk code insertion technique is generally designed to obfuscate malicious OpCode sequences and reduce the ‘proportion’ of malicious OpCodes in a malware.

SYSTEM DESIGN

1. ARCHITECTURE DIAGRAM

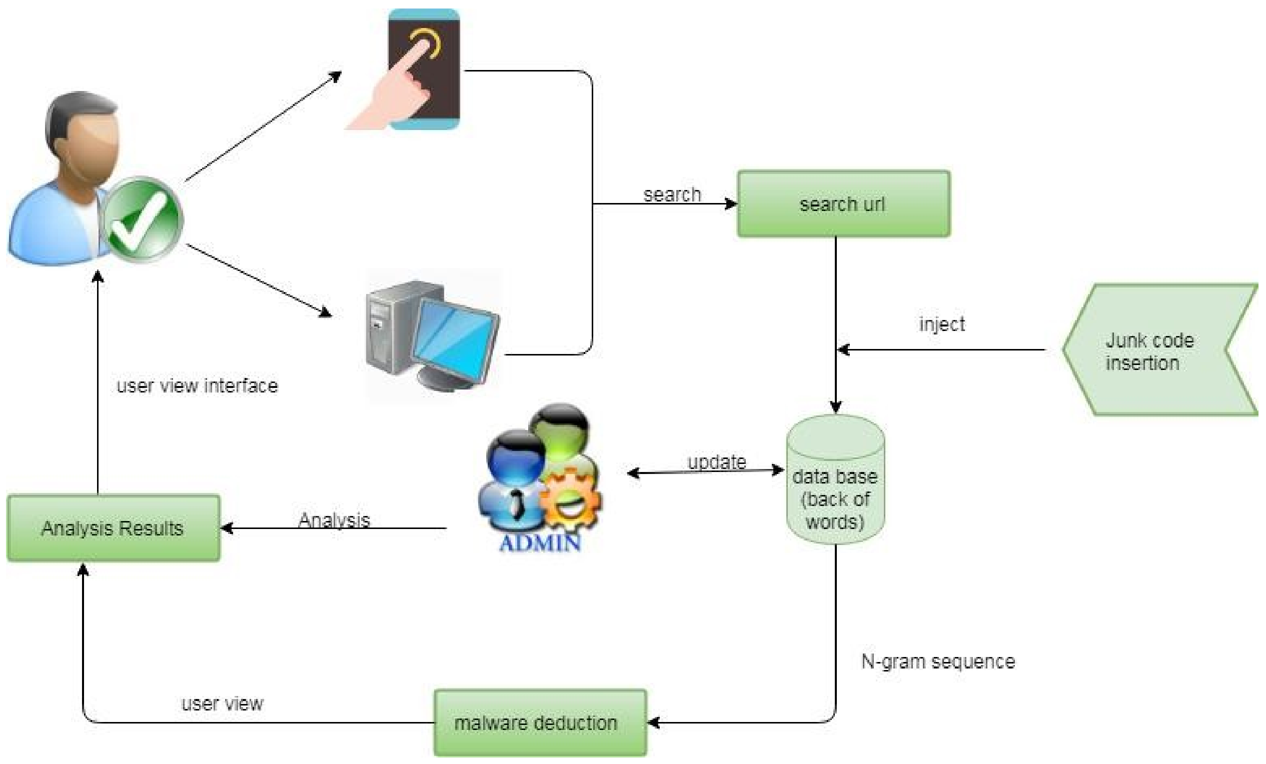


Fig 1: ARCHITECTURE DIAGRAM

1. COMPONENT DIAGRAM

a. User

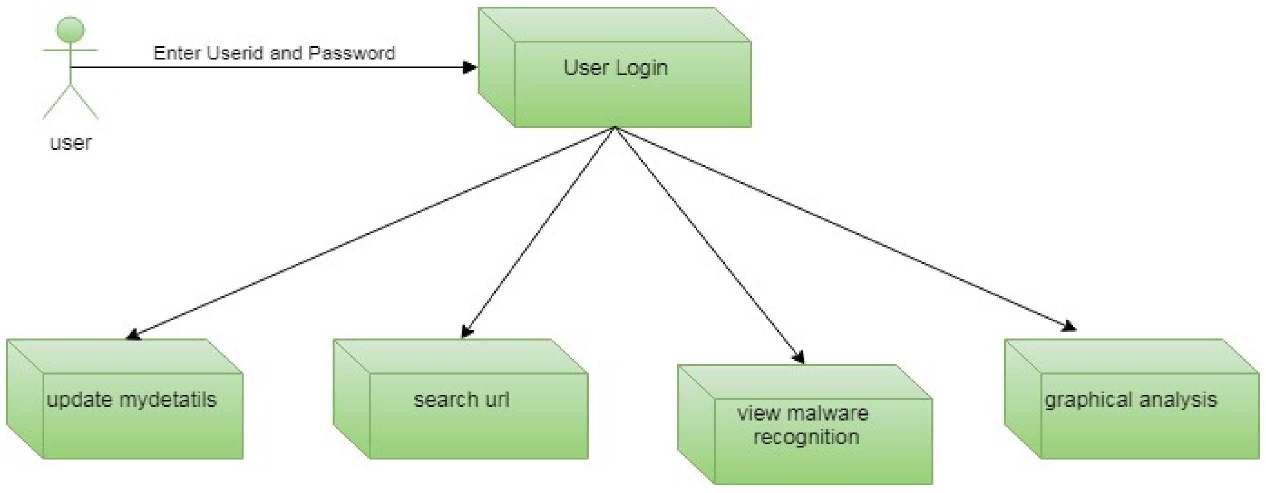


Fig 2: COMPONENT DIAGRAM(user)

a. Admin

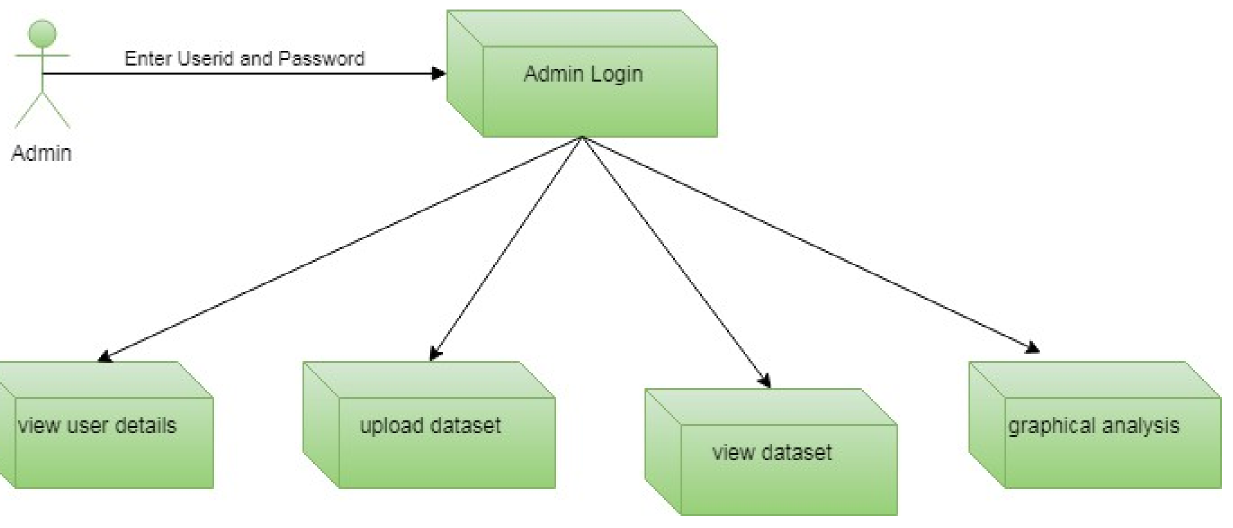


Fig 3: COMPONENT DIAGRAM (admin)

2. ER DIAGRAM

a. User

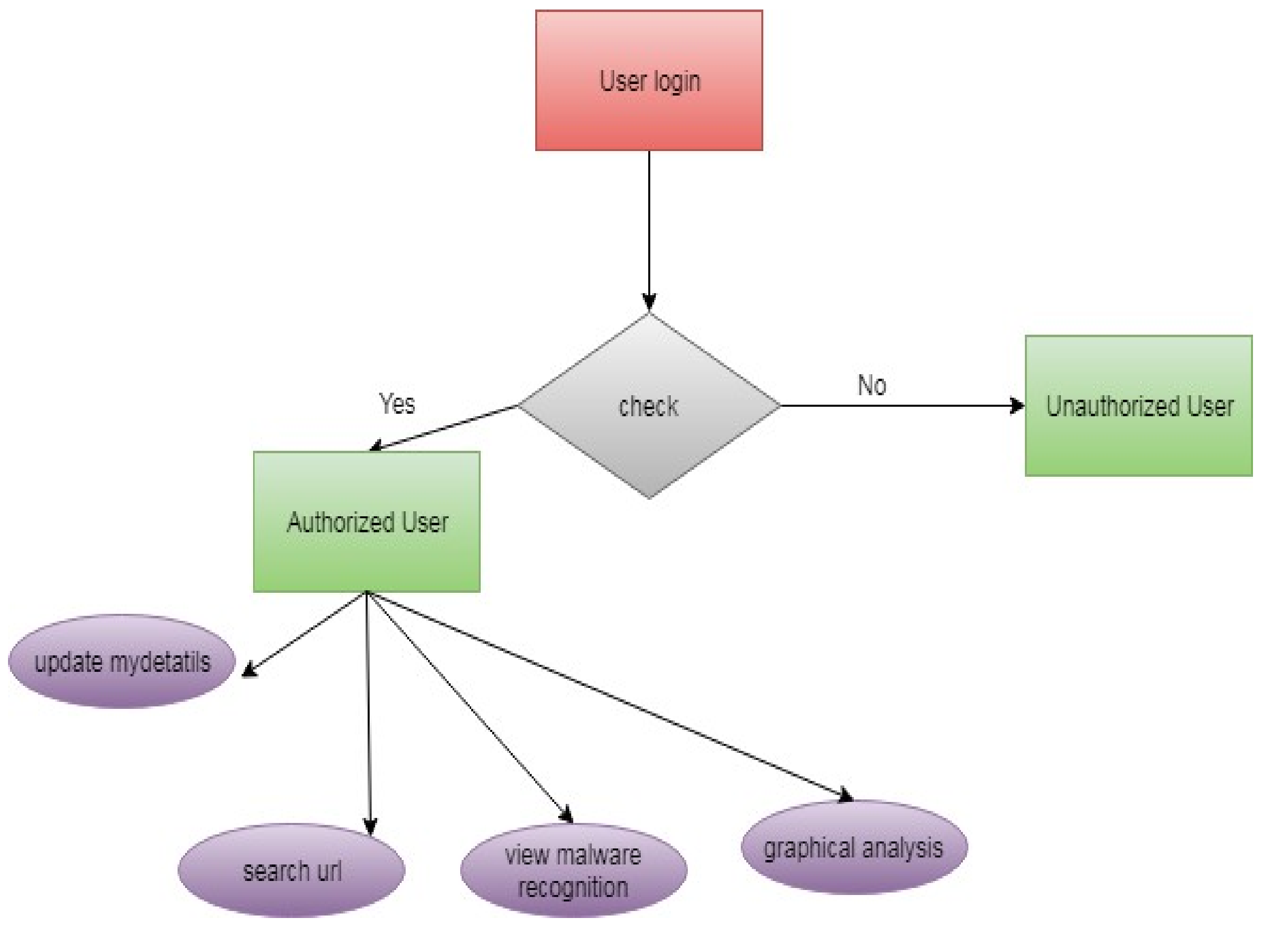


Fig 4: ER DIAGRAM (user)

a. Admin

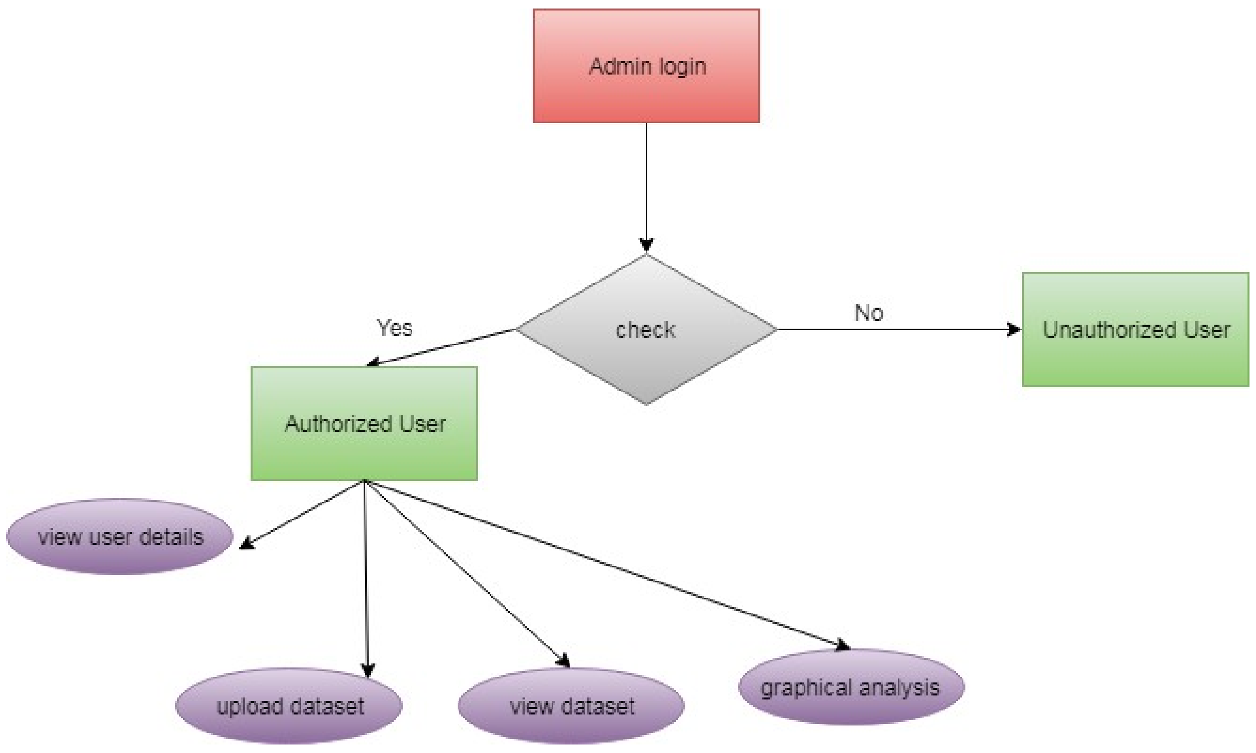


Fig 5: ER DIAGRAM (admin)

2. USE CASE DIAGRAM

a. User

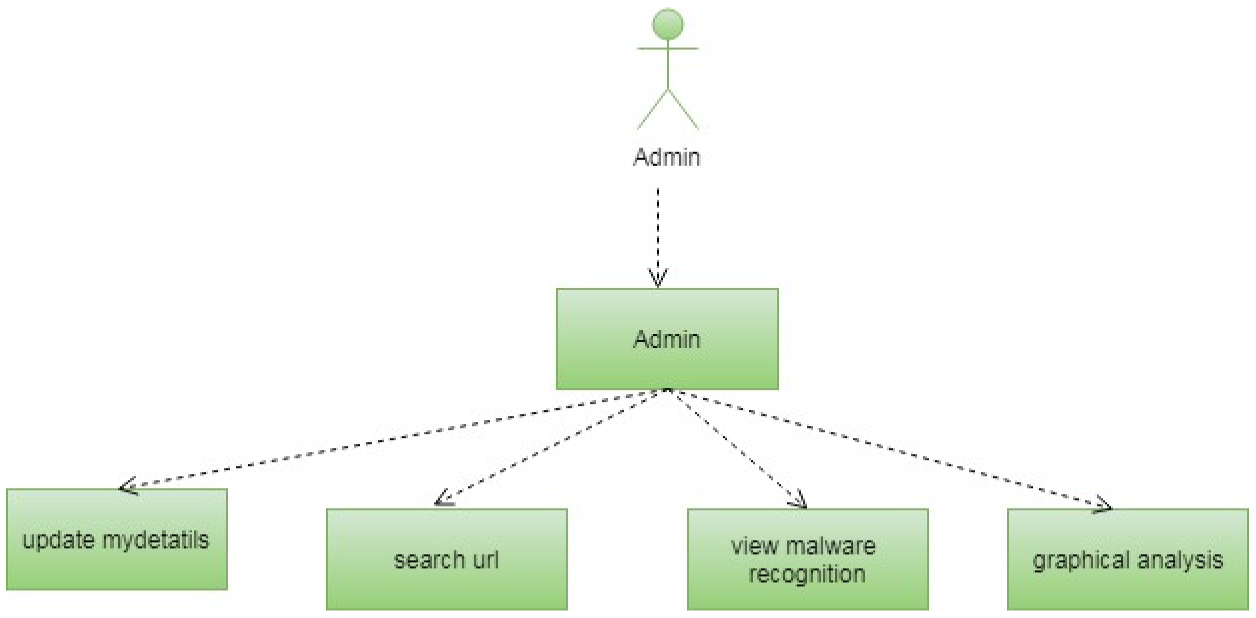


Fig 6: USE CASE DIAGRAM (user)

a. Admin

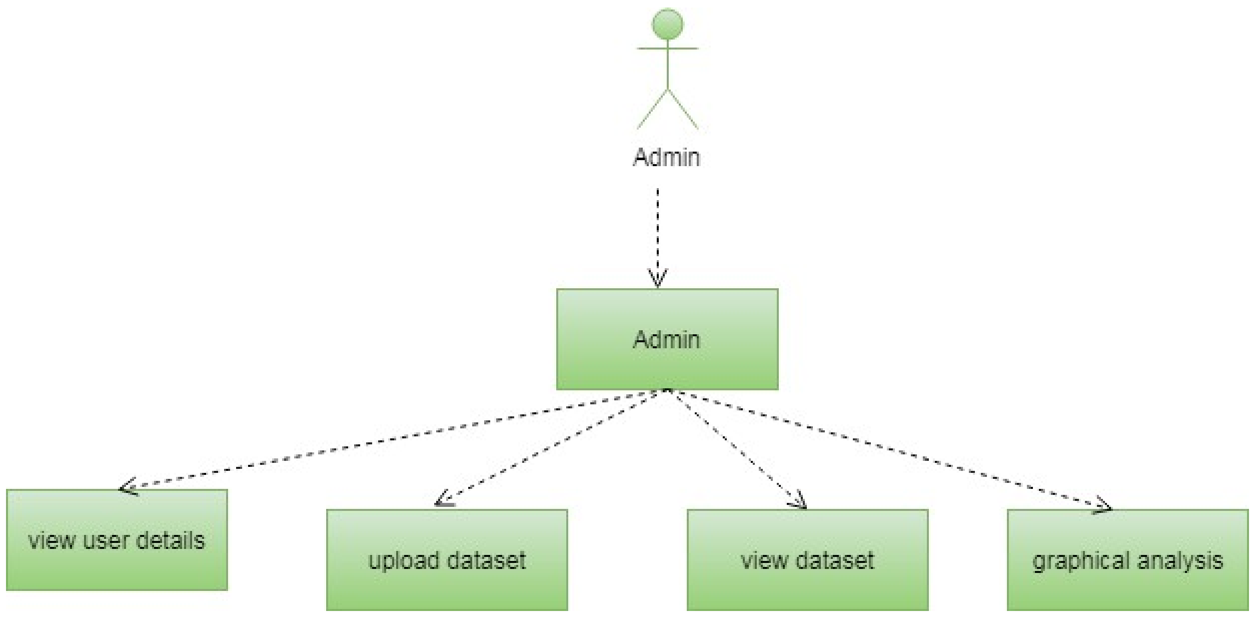


Fig 7: USE CASE DIAGRAM (admin)

2. CLASS DIAGRAM

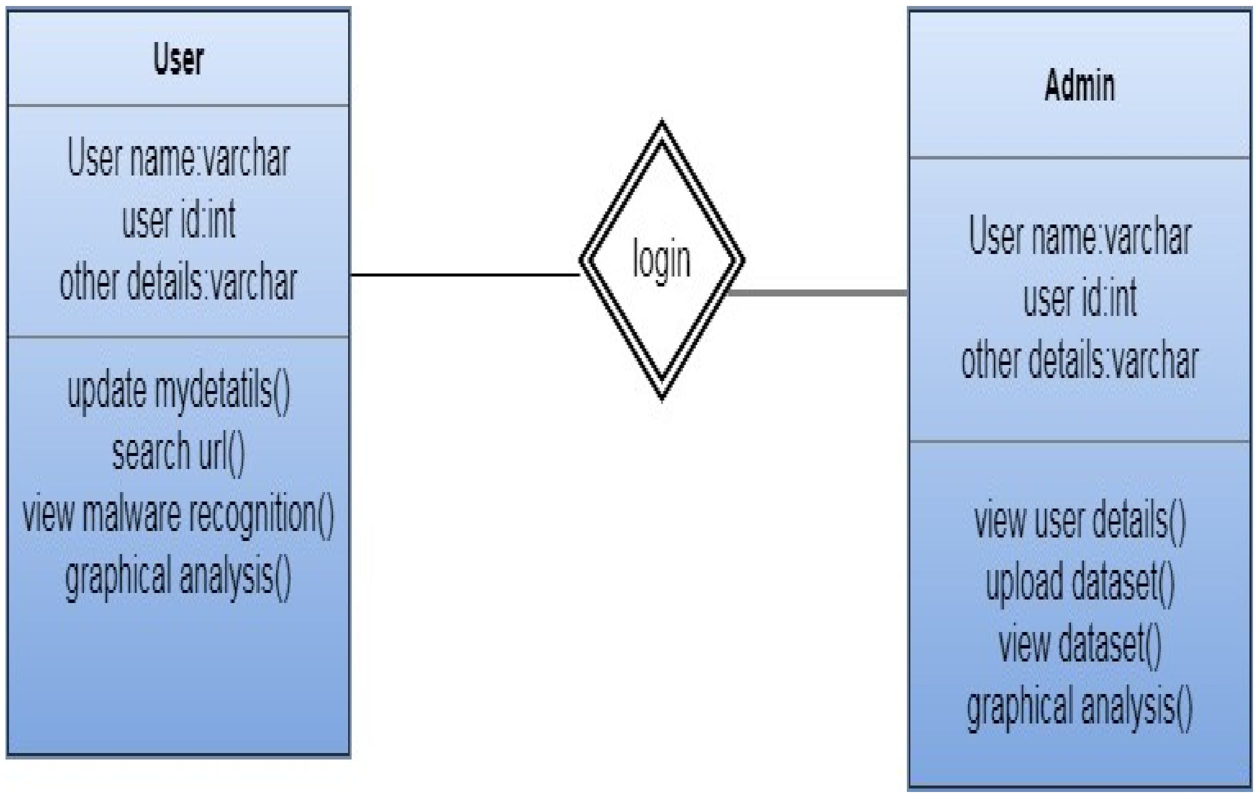


Fig 8: CLASS DIAGRAM

3. DATA FLOW DIAGRAM

a. User

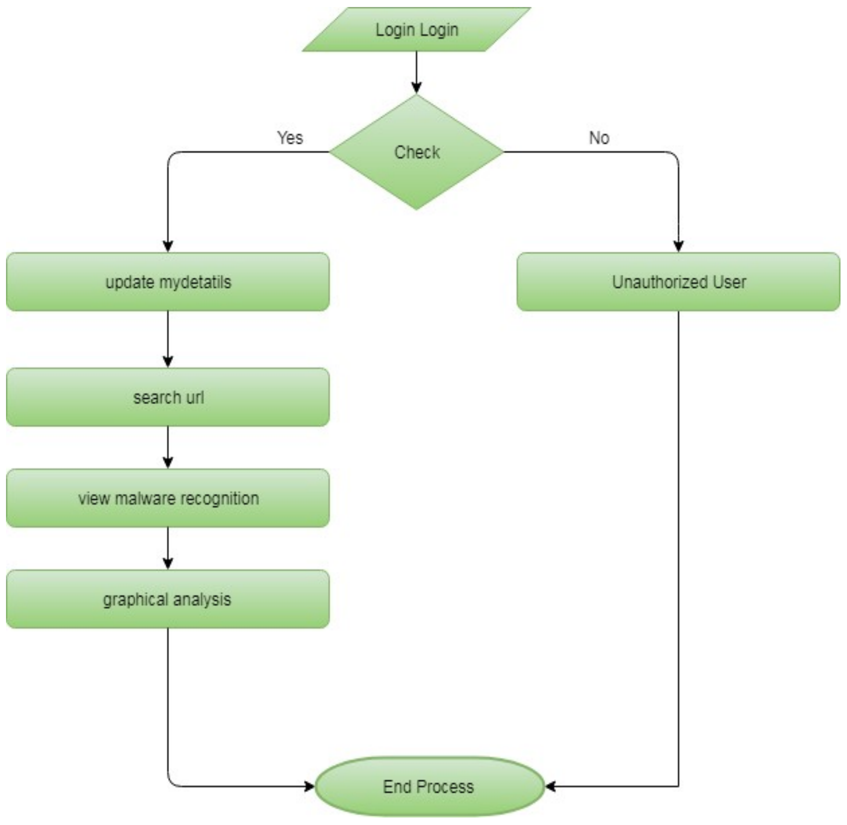


Fig 9: DATA FLOW DIAGRAM (user)

a. Admin

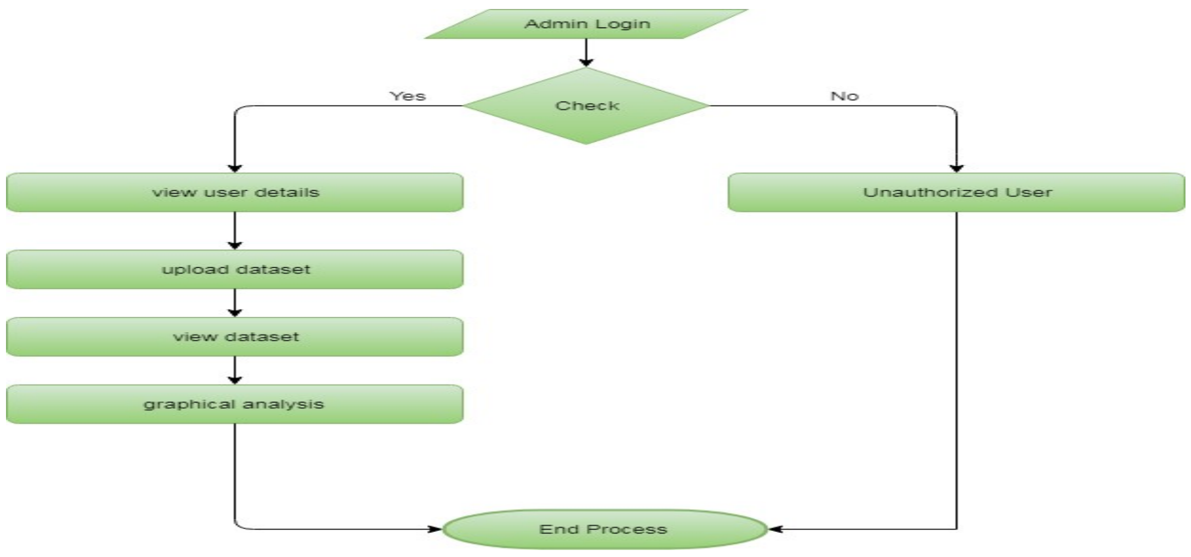


Fig 9: DATA FLOW DIAGRAM (admin)

2. ACTIVITY DIAGRAM

a. User

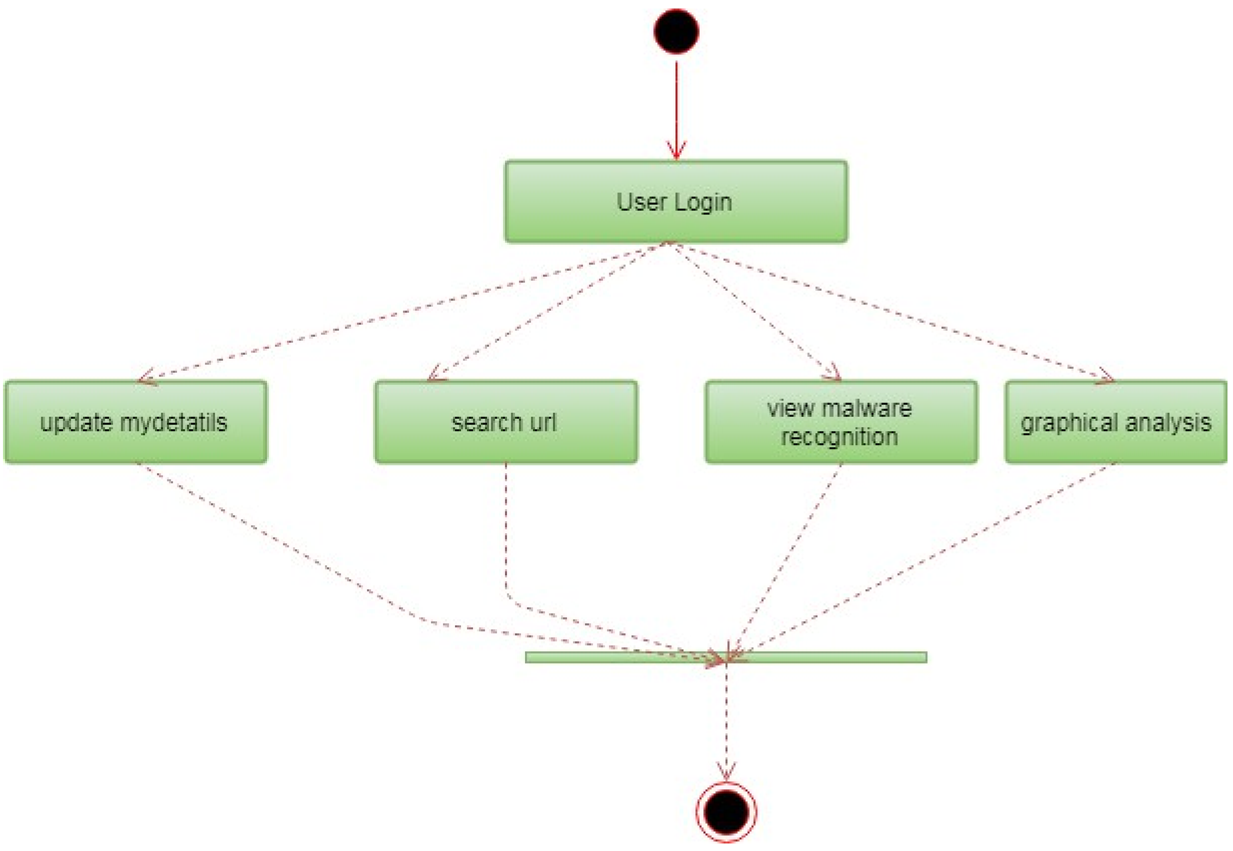


Fig 10: ACTIVITY DIAGRAM (user)

a. Admin

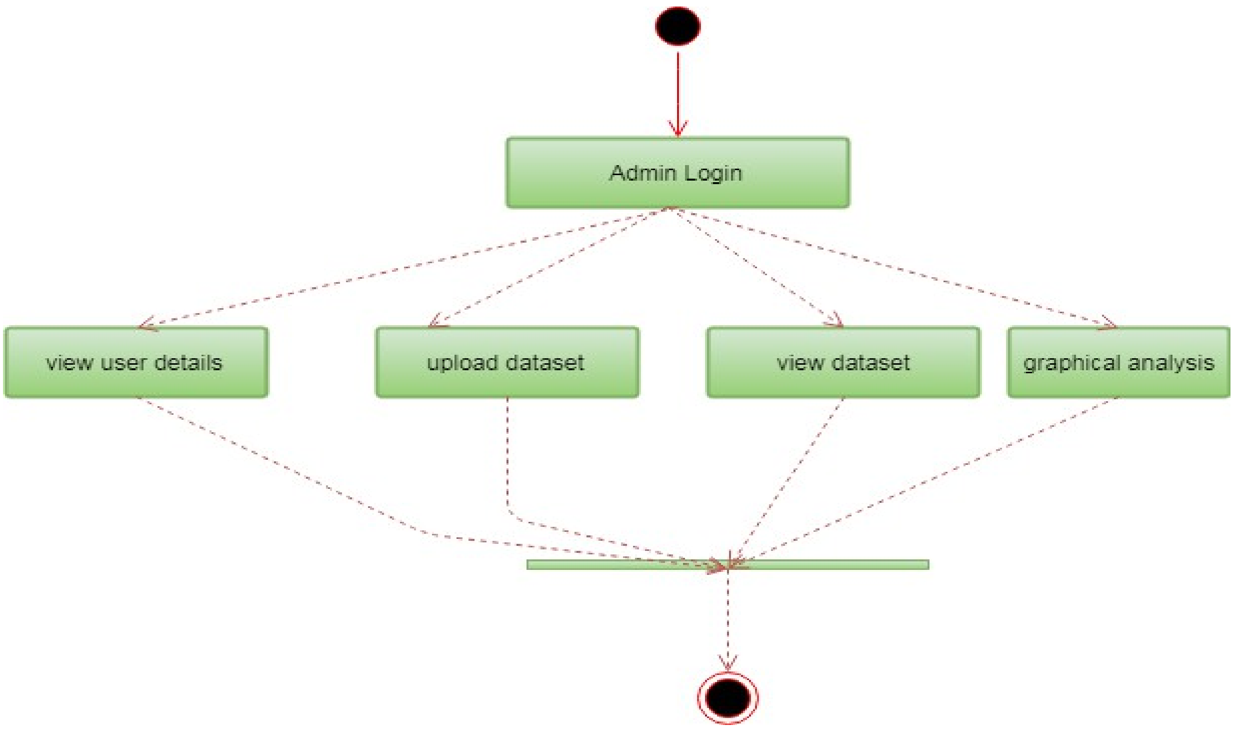


Fig 11: ACTIVITY DIAGRAM (admin)

2. SEQUENCE DIAGRAM

a. User

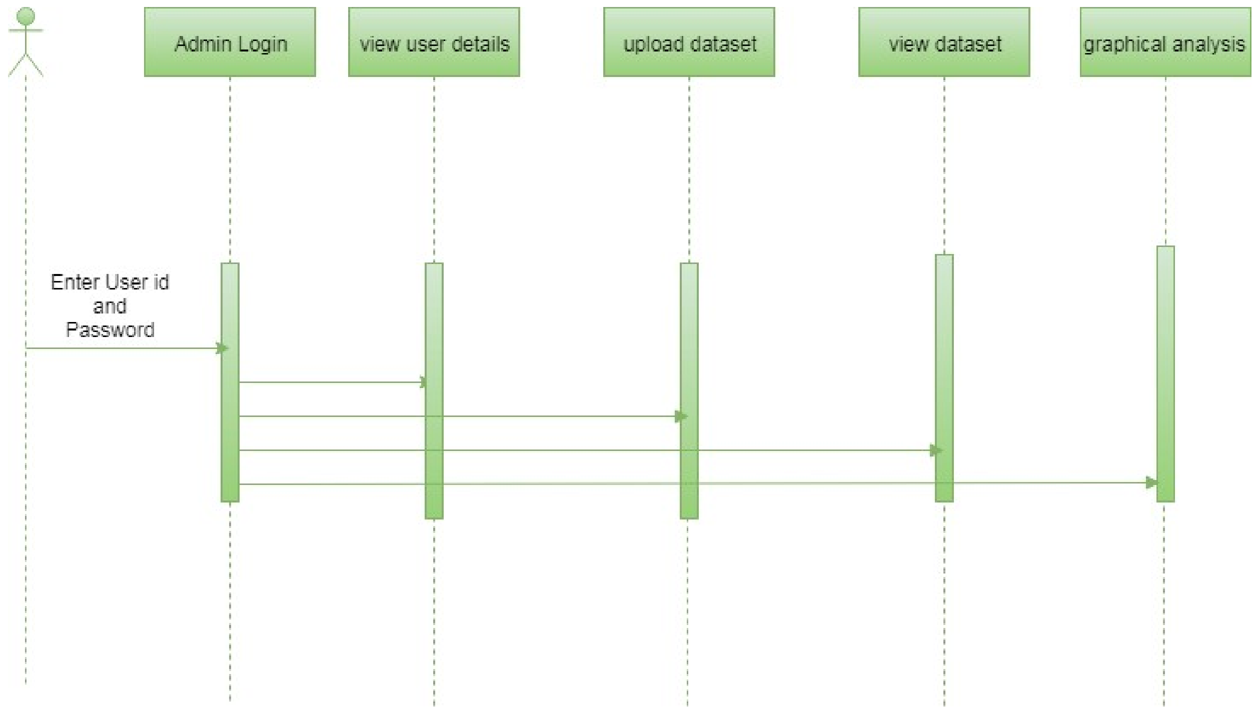


Fig 12: SEQUENCE DIAGRAM (user)

b. Admin

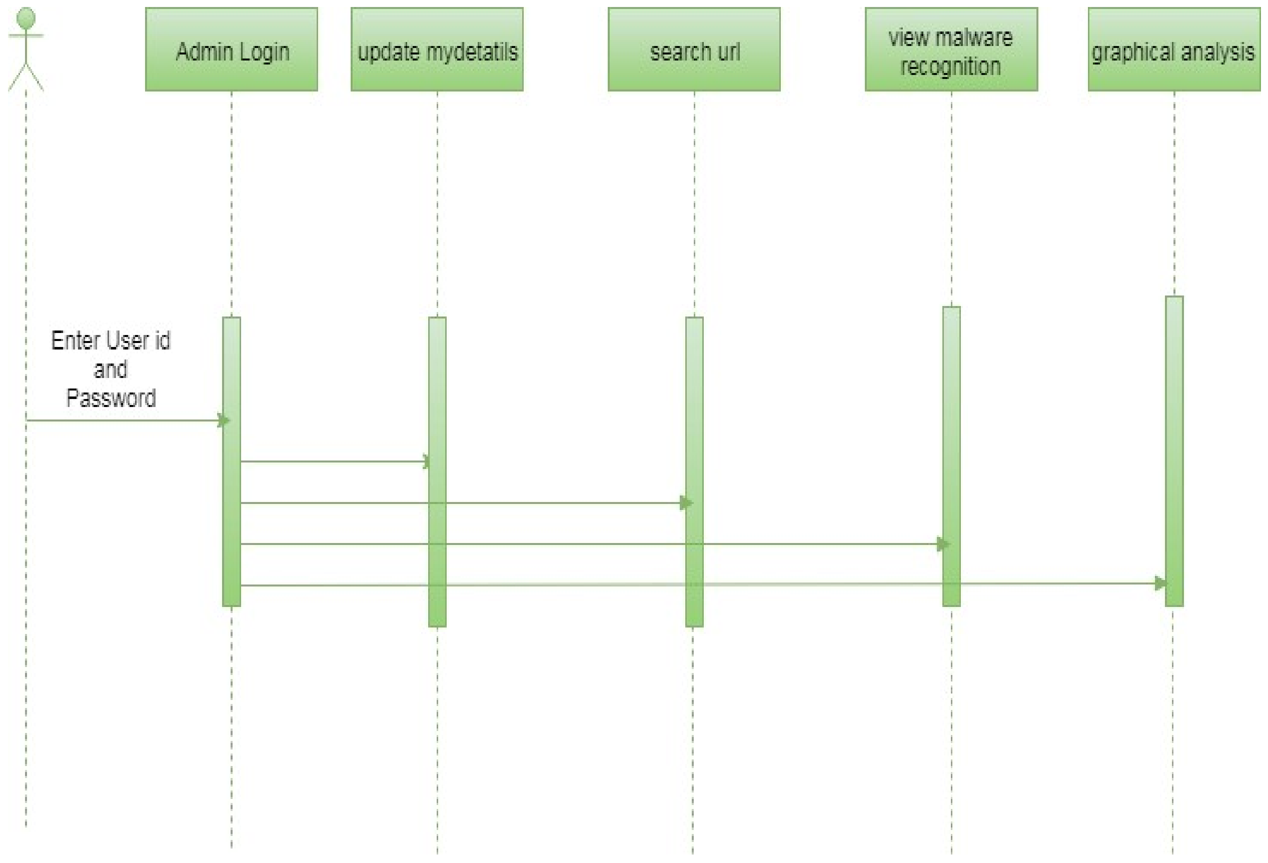


Fig 13: SEQUENCE DIAGRAM (admin)

IMPLEMENTATION

PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library

DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusabilityand "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.

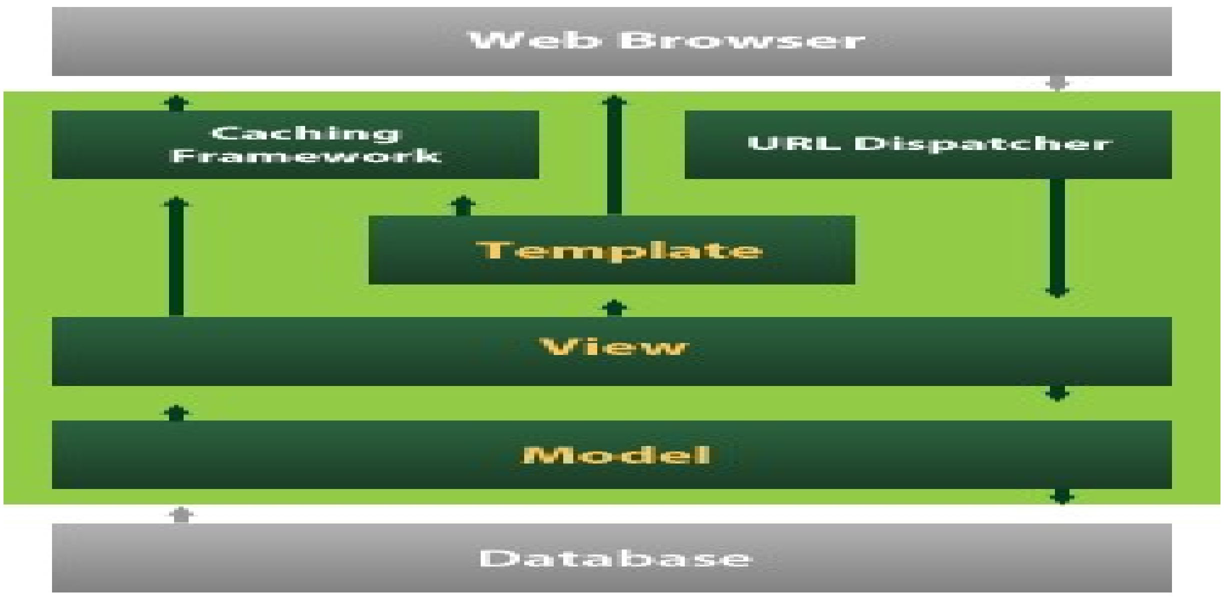


Fig 14:django

Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models

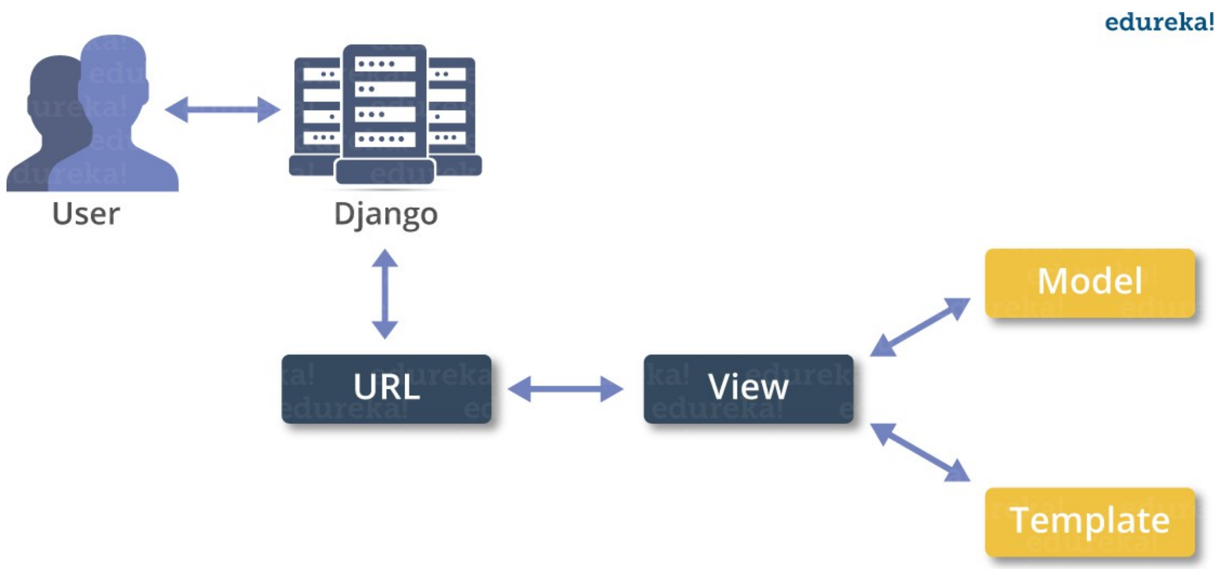


Fig 15:working of django

SMAPLE CODE

```
<!DOCTYPE html>

{% load staticfiles %}

<html lang="en">

        <head>

                <meta charset="utf-8">

                <meta http-equiv="X-UA-Compatible" content="IE=edge">

                <meta name="viewport" content="width=device-width, initial-scale=1">

                <!-- The above 3 meta tags *must* come first in the head; any other head content must
come *after* these tags -->

                <meta name="description" content="">

                <meta name="author" content="">

                <link rel="icon" href="static/favicon.ico">

                <title></title>

                <!-- Bootstrap core CSS -->

                <link href="static/css/bootstrap.min.css" rel="stylesheet">

                <link rel="stylesheet" href="static/https://maxcdn.bootstrapcdn.com/font-
awesome/4.4.0/css/font-awesome.min.css">

                <!-- Custom styles for this template -->

                <link href="static/css/owl.carousel.css" rel="stylesheet">

                <link href="static/css/owl.theme.default.min.css"  rel="stylesheet">

                <link href="static/css/style.css" rel="stylesheet">

                <!-- Just for debugging purposes. Don't actually copy these 2 lines! -->

                <!--[if lt IE 9]><script src="static/../../assets/js/ie8-responsive-file-
warning.js"></script><![endif]-->

                <script src="static/js/ie-emulation-modes-warning.js"></script>

                <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries --
>

                <!--[if lt IE 9]>
```

```
                    <script src="static/https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
                    <script src="static/https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
                    <![endif]-->
            </head>
    <style>
        .userdetails{ ma
    rgin-top:-20px;
    margin-left:30px;


    height:380px;
    width:750px;
    overflow:scroll;


    }
    .userdetails
    table{ width:50e
    m;
    text-align:center;


    border-spacing:1px;
    background:;
}



.userdetails table tr
th{ background:rgb(0,128,128);
padding:5px;
}
.userdetails table tr td{
```

```
background:rgba(0,128,128,0.7);

padding:5px;

}

.userdetails table tr:hover

td{ background:rgba(0,128,128);

}

.sideimage{ border

-style:solid;

border-width:1px;

height:380px;

width:360px;

 margin-top:-380px;

 margin-left:830px;

 background: url("{% static '16.jpg' %}");

 background-size: 100%100%;

float:left;

}


    </style>

            <body style="background-color:white;">

                        <!-- Navigation -->

                        <nav class="navbar navbar-default navbar-fixed-top">

                                    <div class="container">

                                                <!-- Brand and toggle get grouped for better mobile display -->

                                                <div class="navbar-header page-scroll">

                                                            <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1">

                                                            <span class="sr-only">Toggle navigation</span>
```

```
                                            <span class="icon-bar"></span>
                                            <span class="icon-bar"></span>
                                            <span class="icon-bar"></span>
                                            </button>
                                                    </div>
                                    <!-- Collect the nav links, forms, and other content for toggling -->
                                    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
                                            <ul class="nav navbar-nav navbar-right">
                                                    <li class="hidden">
                                                            <a href=""></a>
                                                    </li>
                                                    <li>
                                                            <a  style="color:WHITE;text-decoration: none;"
href="{% url 'admin_page' %}">USER DETAILS</a>
                                                    </li><li>
                                                            <a  style="color:WHITE;text-decoration: none;"
href="{% url 'admin_networkdata' %}">DATA SET</a>
                                                    </li><li>
                                                            <a  style="color:WHITE;text-decoration: none;"
href="{% url 'admin_nlpanalysis' %}">OPCODE BAASED MALWARE</a>
                                                    </li><li>
                                                            <a  style="color:WHITE;text-decoration: none;"
href="{% url 'admin_graphicalanalysis' %}"> GRAPHICAL ANALYSIS
                                                                    </a>
                                                    </li><li>
                                                            <a  style="color:WHITE;text-decoration: none;"
href="{% url 'admin_algorithms' %}">ACCURACY ANALYSIS</a>
                                                    </li><li>
                                                            <a  style="color:WHITE;text-decoration: none;"
href="{% url 'view_feedback' %}">VIEW FEEDBACK</a>
```

```
                                        </li><li>
                                             <a  style="color:WHITE;text-decoration: none;"
href="{% url 'admin_login' %}"> LOGOUT</a>
                                        </li>


                                   </ul>
                              </div>
                              <!-- /.navbar-collapse -->
                         </div>
                         <!-- /.container-fluid -->
                    </nav>
                    <!-- Header -->
                    <header>
                         <div class="container">
                              <div class="slider-container">
                                   <div class="intro-text">
                                        <div class="intro-lead-in" style="color:lime">Robust
Malware Detection for Internet Of
(Battlefield) Things Devices Using Deep
Eigenspace Learning</div>



                                   </div>
                              </div>
                         </div>
                    </header>
                    <section id="about" class="light-bg">

                         <div class="userdetails">
```

```
<table>
   <tr>
      <th style="color:white"> FIRST NAME</th>
      <th style="color:white">LAST NAME</th>
      <th style="color:white">USER ID</th>
      <th style="color:white">MOBILE NUMBER</th>
      <th style="color:white"><label style="margin-left:60px;">EMAIL</label></th>
      <th style="color:white">GENDER</th>
   </tr>
   {% for i in obje %}
   <tr>
   <td style="color:white">{{i.firstname}}</td>
   <td style="color:white">{{i.lastname}}</td>
   <td style="color:white">{{i.userid}}</td>
   <td style="color:white">{{i.mblenum}}</td>
   <td style="color:white">{{i.email}}</td>
   <td style="color:white">{{i.gender}}</td>
</tr>
   {% endfor %}
</table>
</div>
<div class="sideimage"></div>


                  </section>



                  <footer>
```

```
                              <div class="container text-center">
            <marquee style="color:yellow">Robust Malware Detection for Internet Of
(Battlefield) Things Devices Using Deep
Eigenspace Learning</marquee>
                              </div>
                    </footer>


                    <!-- Modal for portfolio item 1 -->
                    <div class="modal fade" id="Modal-1" tabindex="-1" role="dialog" aria-
labelledby="Modal-label-1">
                              <div class="modal-dialog" role="document">
                                        <div class="modal-content">
                                                  <div class="modal-header">


                                                  </div>
                                                  <div class="modal-body">
                                                            <img src="static/images/demo/portfolio-1.jpg" alt="img01"
class="img-responsive" />


                                                  </div>


                                        </div>
                              </div>
                    </div>


                    <!-- Modal for portfolio item 2 -->
                    <div class="modal fade" id="Modal-2" tabindex="-1" role="dialog" aria-
labelledby="Modal-label-2">
                              <div class="modal-dialog" role="document">
```

```
                                        <div class="modal-content">

                                                <

                                                <div class="modal-body">

                                                        <img src="static/images/demo/portfolio-2.jpg" alt="img01"
class="img-responsive" />




                                        </div>

                                </div>

                        </div>


                        <!-- Modal for portfolio item 3 -->

                        <div class="modal fade" id="Modal-3" tabindex="-1" role="dialog" aria-
labelledby="Modal-label-3">

                                <div class="modal-dialog" role="document">

                                        <div class="modal-content">



                                                <div class="modal-body">

                                                        <img src="static/images/demo/portfolio-3.jpg" alt="img01"
class="img-responsive" />




                                        </div>

                                </div>

                        </div>


                        <!-- Modal for portfolio item 4 -->

                        <div class="modal fade" id="Modal-4" tabindex="-1" role="dialog" aria-
labelledby="Modal-label-4">
```

```
                <div class="modal-dialog" role="document">

                    <div class="modal-content">




                        <div class="modal-footer">

                            <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>

                        </div>

                    </div>

                </div>

            </div>


            <!-- Modal for portfolio item 5 -->

            <div class="modal fade" id="Modal-5" tabindex="-1" role="dialog" aria-
labelledby="Modal-label-5">

                <div class="modal-dialog" role="document">

                    <div class="modal-content">


                        <div class="modal-footer">

                            <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>

                        </div>

                    </div>

                </div>

            </div>


            <!-- Bootstrap core JavaScript

                ================================================== -->

            <!-- Placed at the end of the document so the pages load faster -->
```

```
                    <script
src="static/https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

                    <script src="static/http://cdnjs.cloudflare.com/ajax/libs/jquery-
easing/1.3/jquery.easing.min.js"></script>

                    <script src="static/js/bootstrap.min.js"></script>

                    <script src="static/js/owl.carousel.min.js"></script>

                    <script src="static/js/cbpAnimatedHeader.js"></script>

                    <script src="static/js/theme-scripts.js"></script>

                    <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->

                    <script src="static/js/ie10-viewport-bug-workaround.js"></script>

          </body>

</html>

<!DOCTYPE html>

{% load staticfiles %}

<html lang="en">

          <head>

                    <meta charset="utf-8">

                    <meta http-equiv="X-UA-Compatible" content="IE=edge">

                    <meta name="viewport" content="width=device-width, initial-scale=1">

                    <!-- The above 3 meta tags *must* come first in the head; any other head content must
come *after* these tags -->

                    <meta name="description" content="">

                    <meta name="author" content="">

                    <link rel="icon" href="static/favicon.ico">

                    <title></title>

                    <!-- Bootstrap core CSS -->

                    <link href="../static/css/bootstrap.min.css" rel="stylesheet">

                    <link rel="../stylesheet" href="static/https://maxcdn.bootstrapcdn.com/font-
awesome/4.4.0/css/font-awesome.min.css">
```

```
                        <!-- Custom styles for this template -->
                        <link href="../static/css/owl.carousel.css" rel="stylesheet">
                        <link href="../static/css/owl.theme.default.min.css"  rel="stylesheet">
                        <link href="../static/css/style.css" rel="stylesheet">
                        <!-- Just for debugging purposes. Don't actually copy these 2 lines! -->
                        <!--[if lt IE 9]><script src="static/../../assets/js/ie8-responsive-file-
warning.js"></script><![endif]-->
                        <script src="../static/js/ie-emulation-modes-warning.js"></script>
                        <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries --
>
                        <!--[if lt IE 9]>
                        <script src="../static/https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
                        <script src="../static/https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
                        <![endif]-->
            </head>
    <style>
       .updatedetails{ p
    osition: absolute;
    margin-top:-40px;
    left:130px;
    padding:10px;
    width:500px;
}

.updatedetails
    table{ width:30e
    m;
    text-align:center;
    //border-collapse:collapse;
    border-spacing:1px;
    background:;
}


.updatedetails table tr
    th{ color:;
}
.updatedetails table tr
th{ background:
padding:10px;
}
.updatedetails table tr
td{ background:rgb(0,128,128);
padding:10px;
}
.updatedetails table tr:hover
td{ background:r);
}
.updateimage{ bord
er-style:none;
border-width:1px;
height:350px;
width:350px;
margin-top:-20px;
margin-left:740px;
 background: url("{% static '12.png' %}");
 background-size: 100%100%;


}
```

```
   </style>

        <body style="background-color:white;">

                <!-- Navigation -->

                <nav class="navbar navbar-default navbar-fixed-top">

                        <div class="container">

                                <!-- Brand and toggle get grouped for better mobile display -->

                                <div class="navbar-header page-scroll">

                                        <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1">

                                        <span class="sr-only">Toggle navigation</span>

                                        <span class="icon-bar"></span>

                                        <span class="icon-bar"></span>

                                        <span class="icon-bar"></span>

                                        </button>

                                                </div>

                                <!-- Collect the nav links, forms, and other content for toggling -->

                                <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

                                        <ul class="nav navbar-nav navbar-right">

                                                <li class="hidden">

                                                        <a href=""></a>

                                                </li>

                                                <li>

                                                        <a  style="color:WHITE;text-decoration: none;"
href="{% url 'mydetails' %}">MY DETAILS</a>

                                                </li><li>
                                                        <a style="color:WHITE;text-decoration: none;" 
href="{% url 'update_page' %}">UPDATE DETAILS</a>

                                                </li><li>

                                                        <a style="color:WHITE;text-decoration: none;" 
href="{% url 'userpage' %}">HOME PAGE</a>

                                                </li><li>

                                                        <a style="color:WHITE;text-decoration: none;" 
href="{% url 'user_nlp_analysis' %}">NLP ANALYSIS</a>

                                                </li><li>

                                                        <a style="color:WHITE;text-decoration: none;" 
href="{% url 'user_graphical_analysis' %}">GRAPHICAL ANALYSIS</a>

                                                </li><li>

                                                        <a style="color:WHITE;text-decoration: none;" 
href="{% url 'user_feedback' %}">FEEDBACK</a>

                                                </li><li>

                                                        <a style="color:WHITE;text-decoration: none;" 
href="{% url 'index' %}">LOGOUT</a>

                                                </li>


                                        </ul>

                                </div>

                                <!-- /.navbar-collapse -->

                        </div>

                        <!-- /.container-fluid -->

                </nav>

                <!-- Header -->

                <header>

                        <div class="container">

                                <div class="slider-container">

                                        <div class="intro-text">
```

```
                                                <div class="intro-lead-in" style="color:lime">Robust
Malware Detection for Internet Of
(Battlefield) Things Devices Using Deep
Eigenspace Learning</div>



                                            </div>
                                    </div>
                            </div>
                    </header>
                    <section id="about" class="light-bg">

                            <div class="updatedetails">
<form method="post">
   {% csrf_token %}
   <table>
      <tr>
         <td style="color:white">FIRST NAME</td>
         <td><input type="text" name="FirstName" value="{{obj.firstname}}"></td>
      </tr>
      <tr>

         <td style="color:white">LAST NAME</td>
         <td><input type="text" name="LastName" value="{{obj.lastname}}"></td>
      </tr>
          <tr>
         <td style="color:white">USER ID</td>
         <td><input type="text" name="UserId" value="{{obj.userid}}"></td>
```

```
        </tr>

        <tr>

           <td style="color:white">PASSWORD</td>

           <td><input type="text" name="Password" value="{{obj.password}}"></td>

        </tr>

        <tr>

           <td style="color:white">MOBILE NUMBER</td>

           <td><input type="text" name="MobileNumber" value="{{obj.mblenum}}"></td>

        </tr>

        <tr>

           <td style="color:white"> EMAIL ID</td>

           <td><input type="text" name="EmailId" value="{{obj.email}}"></td>

        </tr>

        <tr>

           <td style="color:white">GENDER</td>

           <td><input type="text" name="Gender" value="{{obj.gender}}"></td>

        </tr>

<tr>

           <td style="text-align:center;" colspan="2"><input type="submit" name="submit"
value="SUBMIT" style="background:red;color:white"></td>

        </tr>

    </table>

</form>


</div>

<div class="updateimage"></div>
```

```
                    </section>




                    <footer>

                              <div class="container text-center">

             <marquee style="color:yellow">Robust Malware Detection for Internet Of

(Battlefield) Things Devices Using Deep

Eigenspace Learning</marquee>

                              </div>

                    </footer>


                    <!-- Modal for portfolio item 1 -->

                    <div class="modal fade" id="Modal-1" tabindex="-1" role="dialog" aria-
labelledby="Modal-label-1">

                              <div class="modal-dialog" role="document">

                                        <div class="modal-content">

                                                  <div class="modal-header">



                                                  </div>

                                                  <div class="modal-body">

                                                            <img src="../static/images/demo/portfolio-1.jpg" alt="img01"
class="img-responsive" />



                                                  </div>



                                        </div>

                              </div>

                    </div>
                    <!-- Modal for portfolio item 2 -->
                    <div class="modal fade" id="Modal-2" tabindex="-1" role="dialog" aria-
labelledby="Modal-label-2">
                              <div class="modal-dialog" role="document">
                                        <div class="modal-content">
                                                  <
                                                  <div class="modal-body">
                                                            <img src="../static/images/demo/portfolio-2.jpg" alt="img01"
class="img-responsive" />



                                        </div>
                              </div>
                    </div>

                    <!-- Modal for portfolio item 3 -->
                    <div class="modal fade" id="Modal-3" tabindex="-1" role="dialog" aria-
labelledby="Modal-label-3">
                              <div class="modal-dialog" role="document">
                                        <div class="modal-content">

                                                  <div class="modal-body">
                                                            <img src="static/images/demo/portfolio-3.jpg" alt="img01"
class="img-responsive" />



                                        </div>
                              </div>
                    </div>
```

```
        <!-- Modal for portfolio item 4 -->

        <div class="modal fade" id="Modal-4" tabindex="-1" role="dialog" aria-
labelledby="Modal-label-4">

            <div class="modal-dialog" role="document">

                <div class="modal-content">



                    <div class="modal-footer">

                        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>

                    </div>

                </div>

            </div>

        </div>


        <!-- Modal for portfolio item 5 -->

        <div class="modal fade" id="Modal-5" tabindex="-1" role="dialog" aria-
labelledby="Modal-label-5">

            <div class="modal-dialog" role="document">

                <div class="modal-content">


                    <div class="modal-footer">

                        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>

                    </div>

                </div>

            </div>

        </div>
```

```
                    <!-- Bootstrap core JavaScript

                              ================================================== -->

                    <!-- Placed at the end of the document so the pages load faster -->

                    <script
src="../static/https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

                    <script src="../static/http://cdnjs.cloudflare.com/ajax/libs/jquery-
easing/1.3/jquery.easing.min.js"></script>

                    <script src="../static/js/bootstrap.min.js"></script>

                    <script src="../static/js/owl.carousel.min.js"></script>

                    <script src="../static/js/cbpAnimatedHeader.js"></script>

                    <script src="../static/js/theme-scripts.js"></script>

                    <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->

                    <script src="../static/js/ie10-viewport-bug-workaround.js"></script>

          </body>

</html>
```

INPUT AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

OUTPUT SCREENS

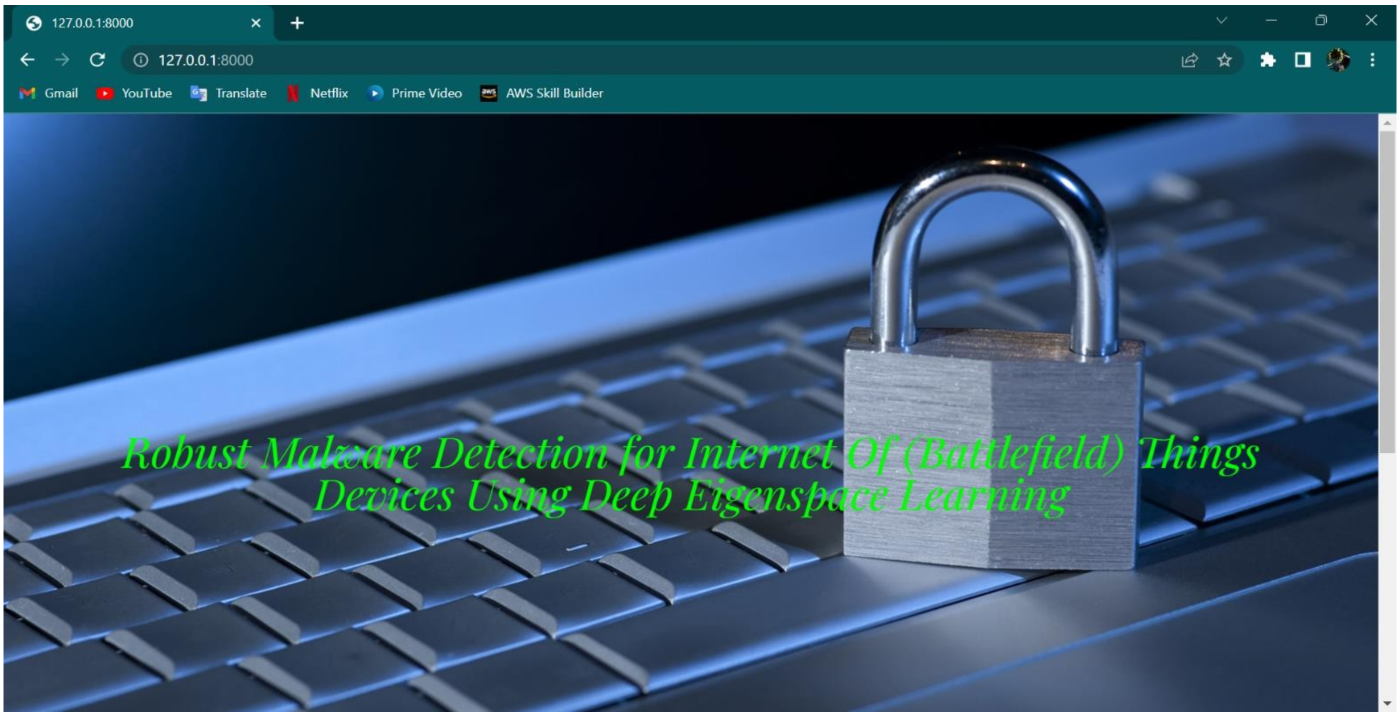

127.0.0.1:8000
Gmail
YouTube
Translate
Netflix
Prime Video
AWS Skill Builder
Robust Malware Detection for Internet Of (Battlefield) Things Devices Using Deep Eigenspace Learning

FIG 16:MAIN HOME PAGE

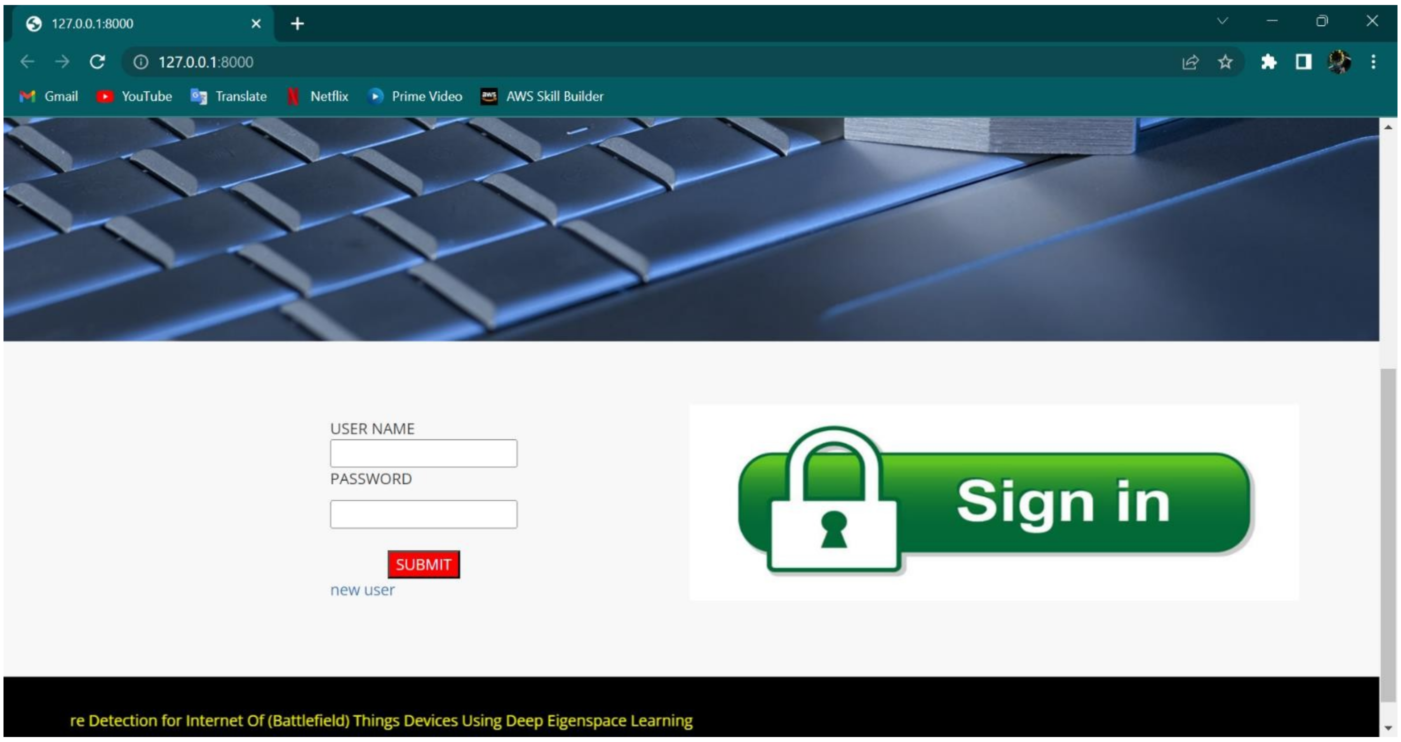
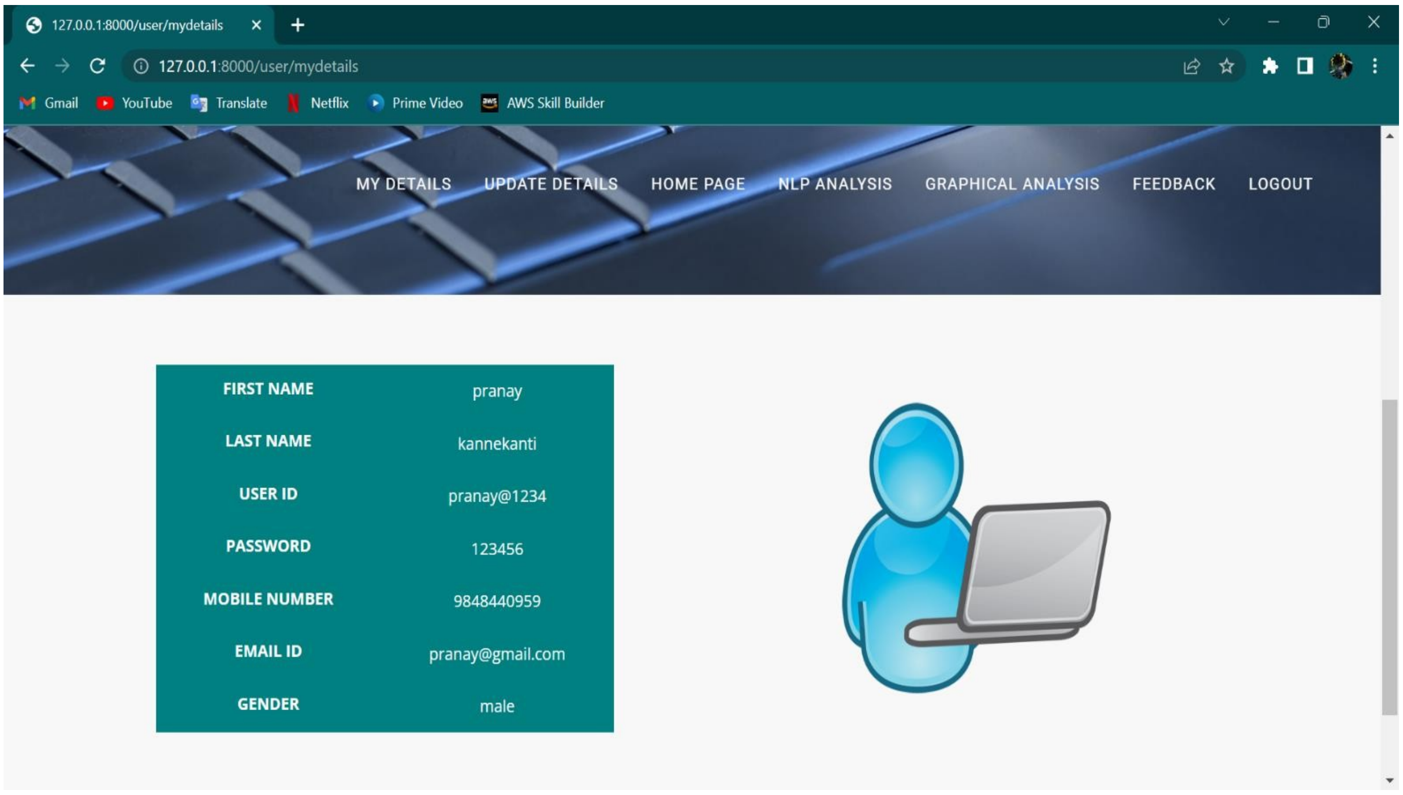
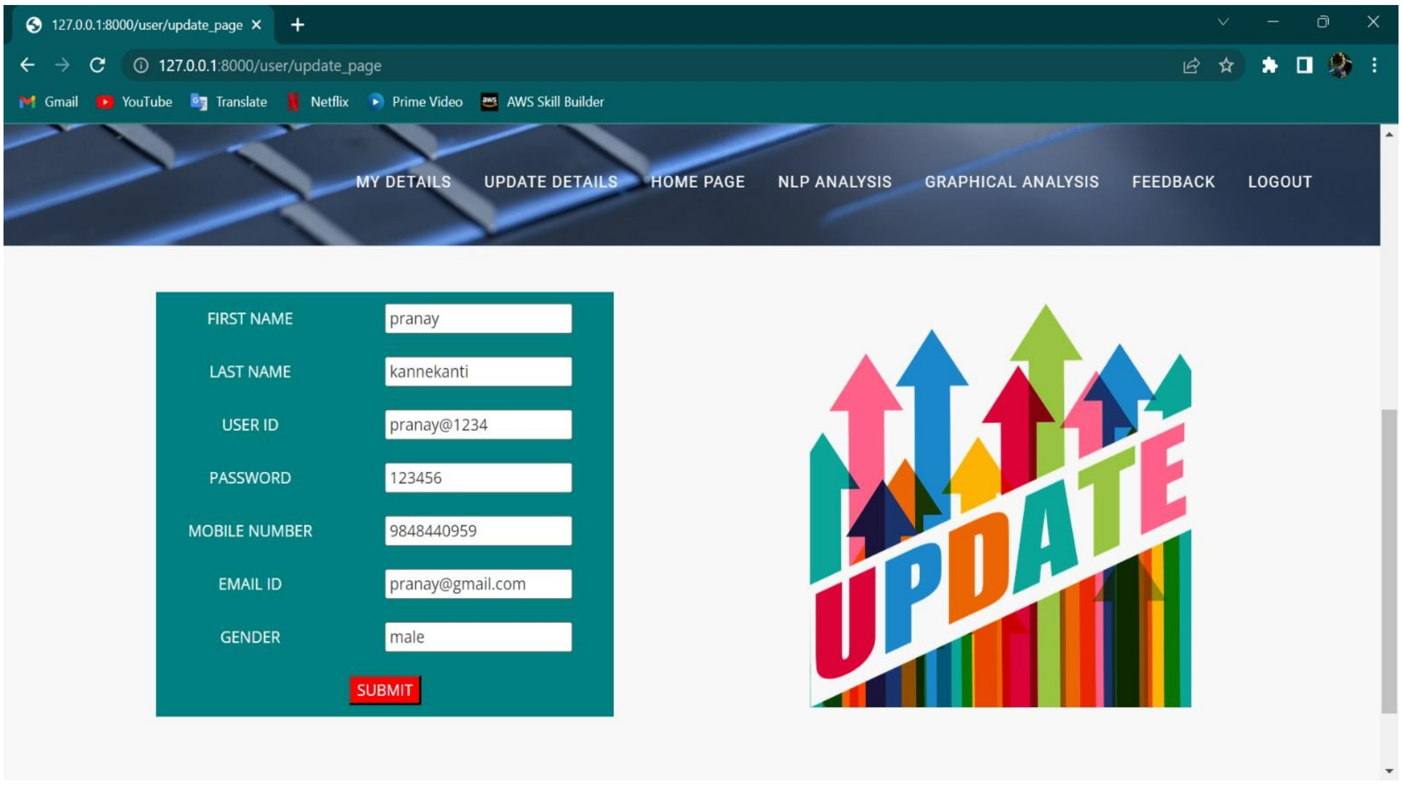


FIG 17: USER LOGIN



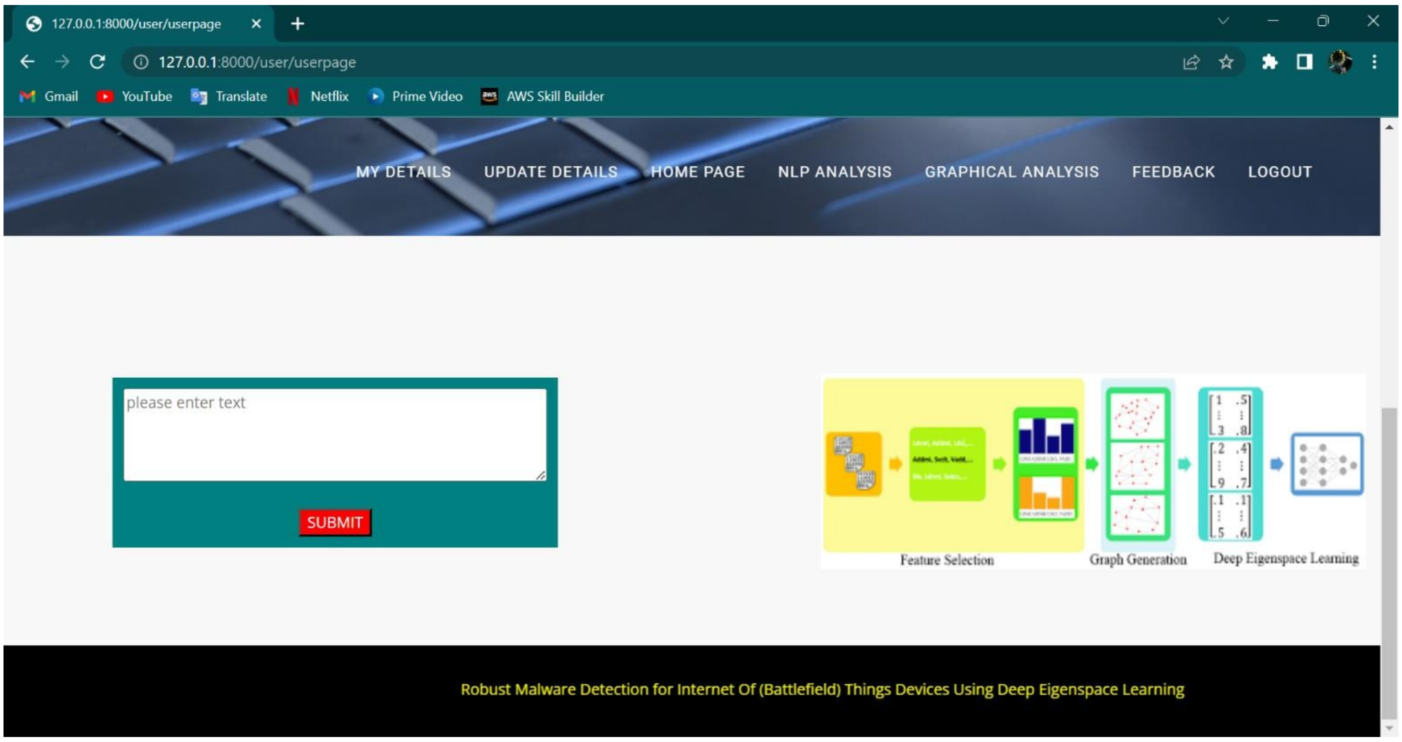
FIRST NAME	pranay
LAST NAME	kannekanti
USER ID	pranay@1234
PASSWORD	123456
MOBILE NUMBER	9848440959
EMAIL ID	pranay@gmail.com
GENDER	male

FIG 18: USER DETAILS



FIRST NAME	pranay
LAST NAME	kannekanti
USER ID	pranay@1234
PASSWORD	123456
MOBILE NUMBER	9848440959
EMAIL ID	pranay@gmail.com
GENDER	male

FIG 19: UPDATE DETAILS


127.0.0.1:8000/user/userpage

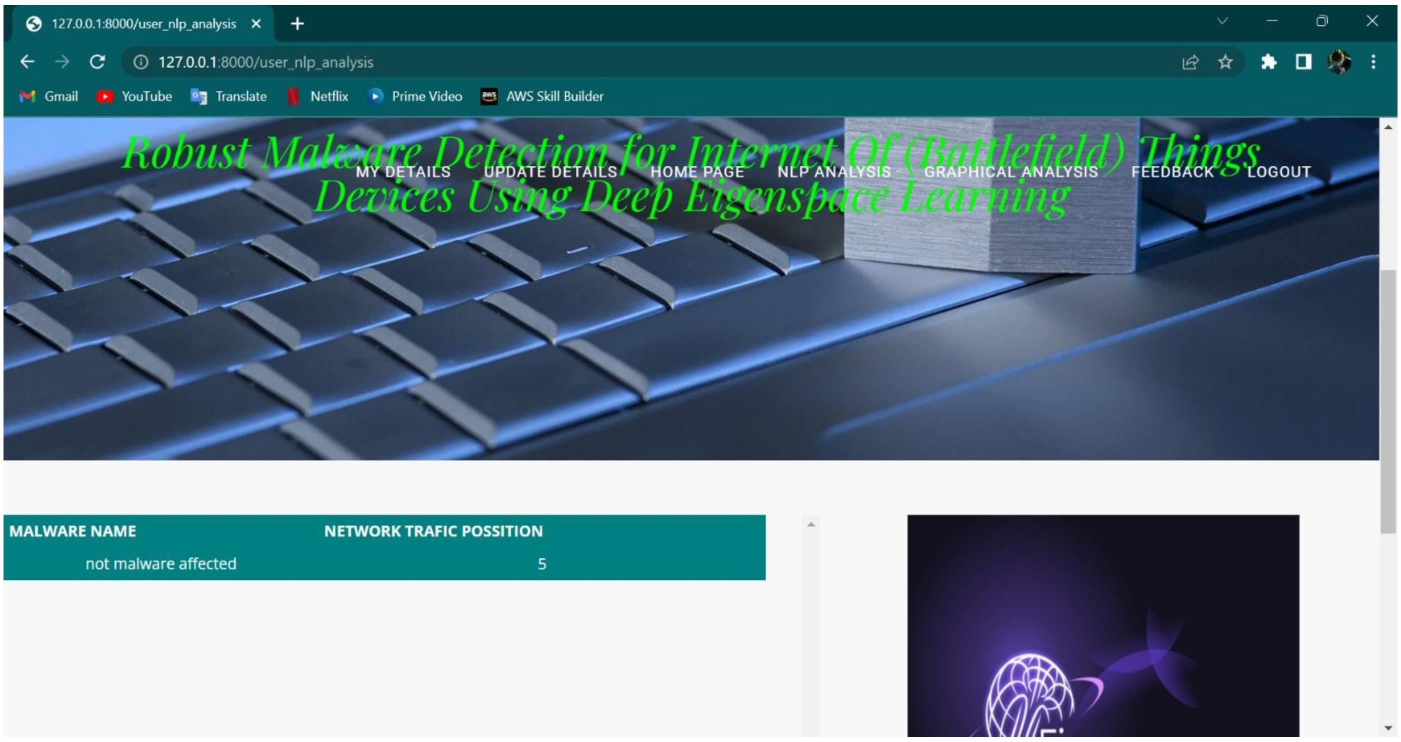
MY DETAILS UPDATE DETAILS HOME PAGE NLP ANALYSIS GRAPHICAL ANALYSIS FEEDBACK LOGOUT

please enter text

SUBMIT

Robust Malware Detection for Internet Of (Battlefield) Things Devices Using Deep Eigenspace Learning

FIG 20: HOME PAGE


127.0.0.1:8000/user_nlp_analysis

Robust Malware Detection for Internet Of (Battlefield) Things Devices Using Deep Eigenspace Learning

MY DETAILS UPDATE DETAILS HOME PAGE NLP ANALYSIS GRAPHICAL ANALYSIS FEEDBACK LOGOUT

MALWARE NAME	NETWORK TRAFIC POSSITION
not malware affected	5

FIG 21: NLP ANALYSIS

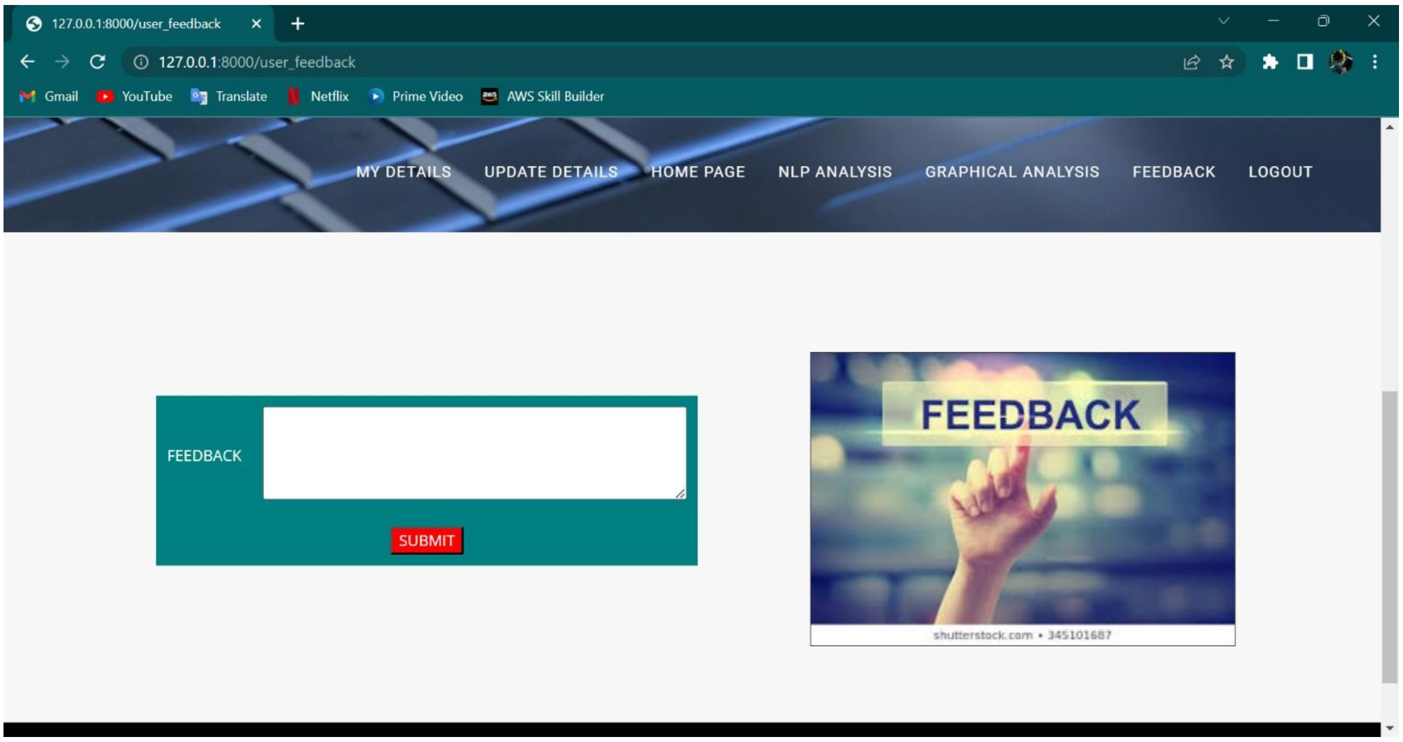


FIG 22: FEEDBACK PAGE

SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CONCLUSION

IoT, particularly IoBT, will be increasingly important in the foreseeable future. No malware detection solution will be foolproof but we can be certain of the constant race between cyber attackers and cyber defenders. Thus, it is important that we maintain persistent pressure on threat actors. In this paper, we presented an IoT and IoBT malware detection approach based on class-wise selection of Op- Codes sequence as a feature for classification task. A graph of selected features was created for each sample and a deep Eigenspace learning approach was used for malware classification. Our evaluations demonstrated the robustness of our approach in malware detection with an accuracy rate of 98.37% and a precision rate of 98.59%, as well as the capability to mitigate junk code insertion attacks.

References

- E. Bertino, K.-K. R. Choo, D. Georgakopolous, and S. Nepal, “Internet of things (iot): Smart and secure service delivery,” ACM Transactions on Internet Technology, vol. 16, no. 4, p. Article No. 22, 2016.
- X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, “A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments,” Journal of Network and Computer Applications, 2017.
- J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” Future generation computer systems, vol. 29, no. 7, pp. 1645– 1660, 2013.
- F. Leu, C. Ko, I. You, K.-K. R. Choo, and C.-L. Ho, “A smartphonebased wearable sensors for monitoring real-time physiological data,” Computers & Electrical Engineering, 2017.