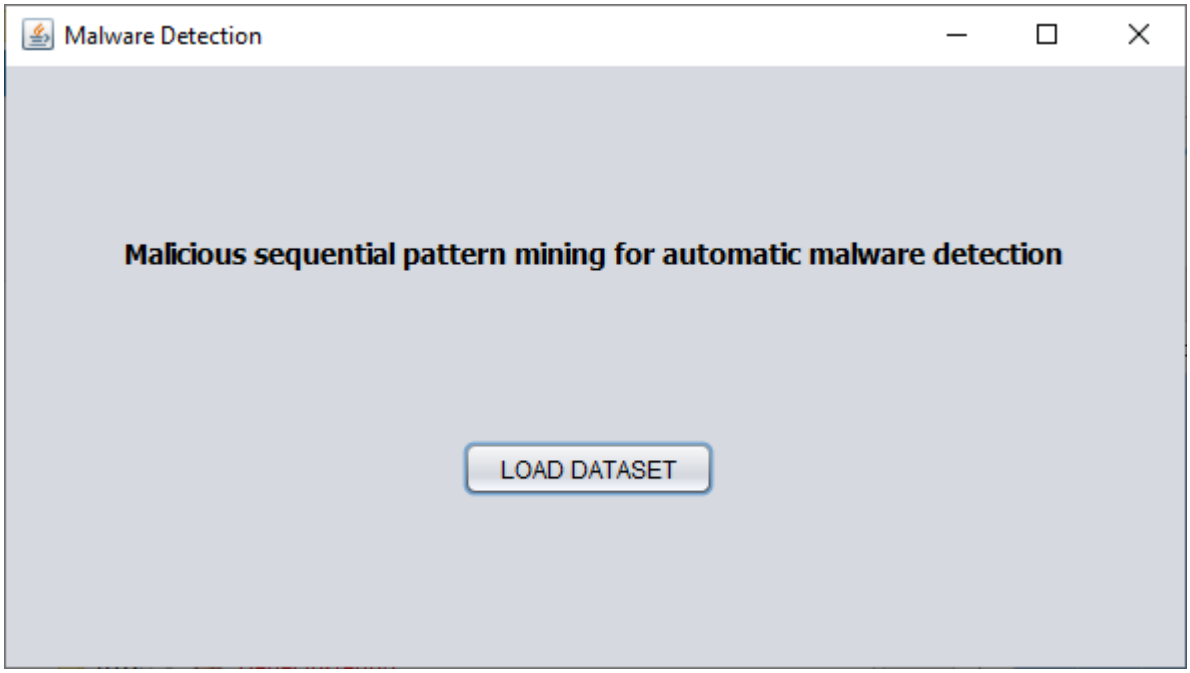
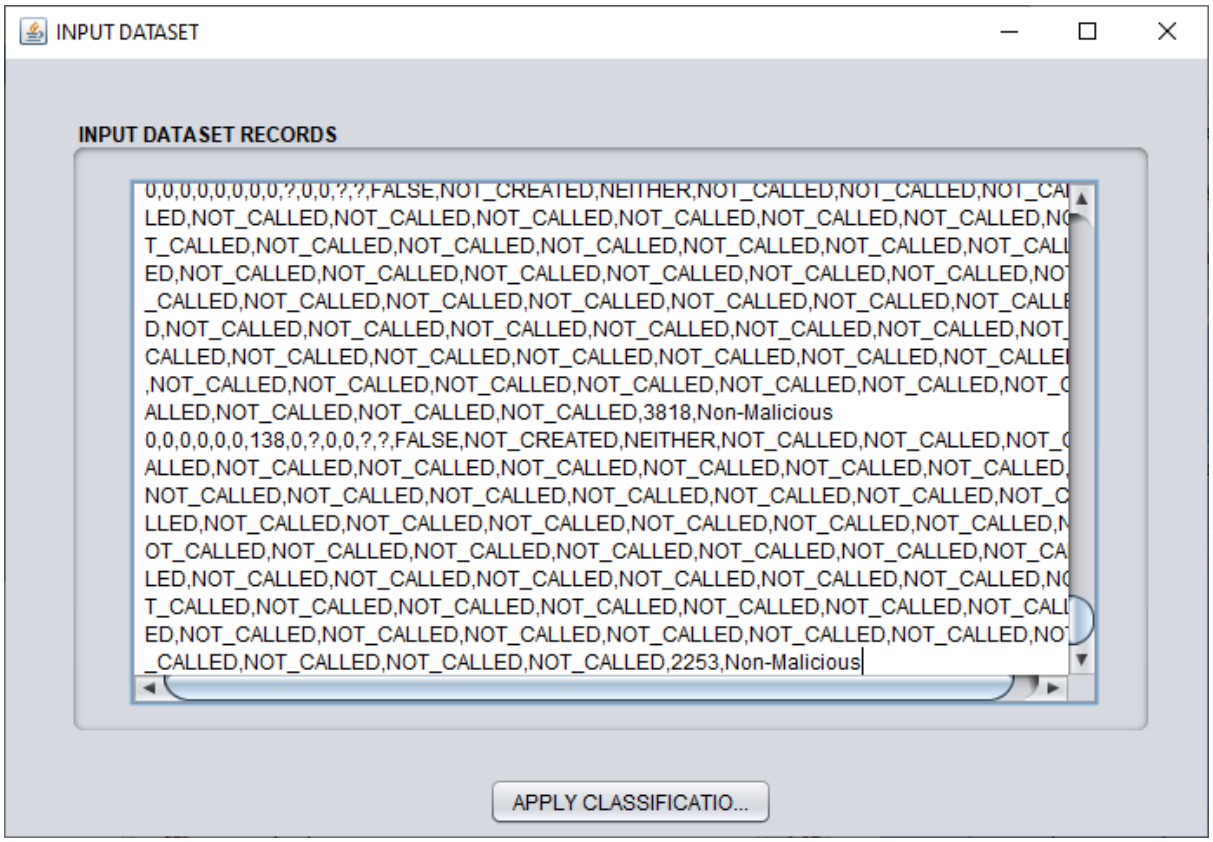
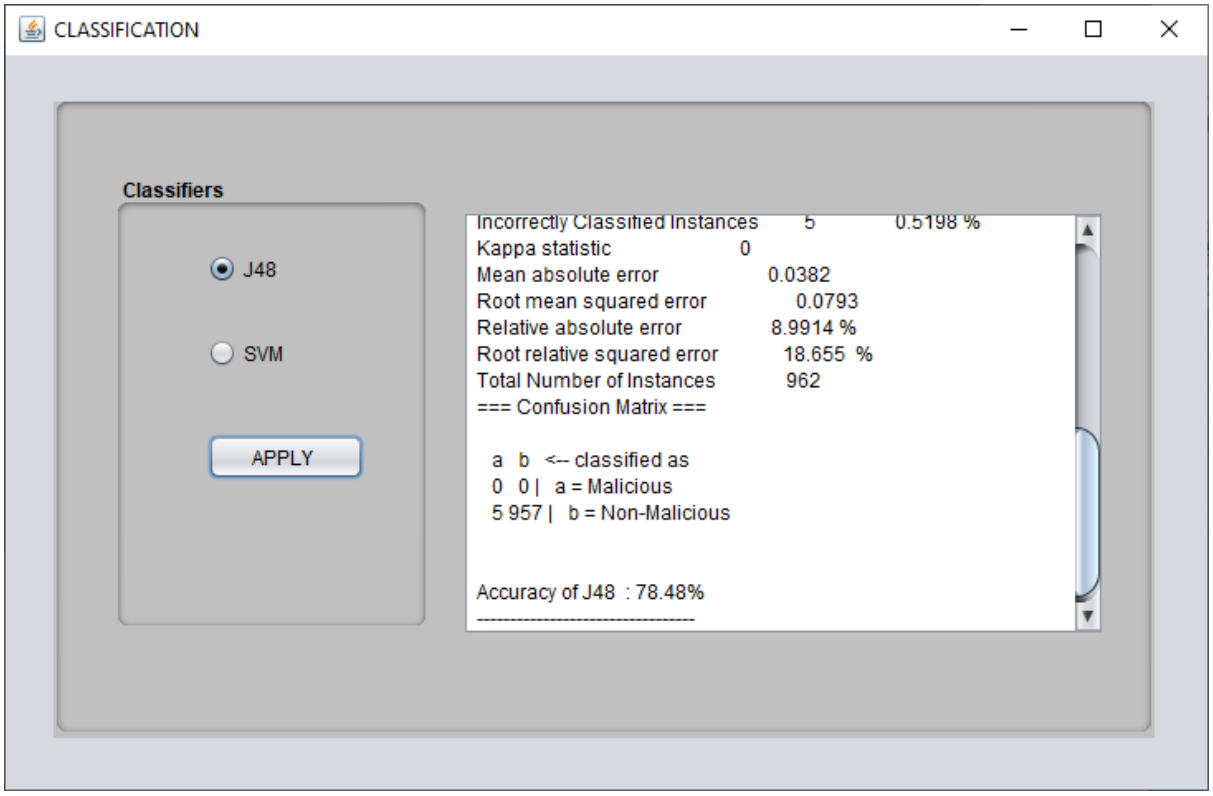
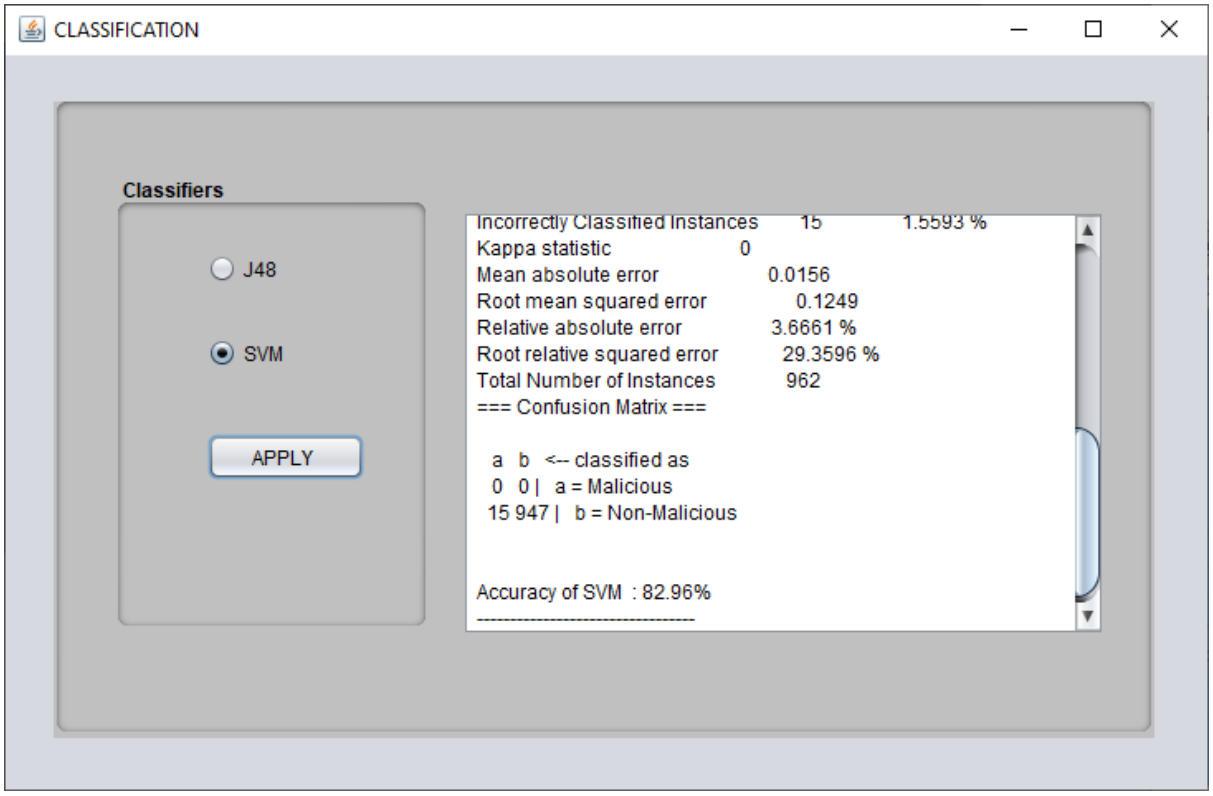
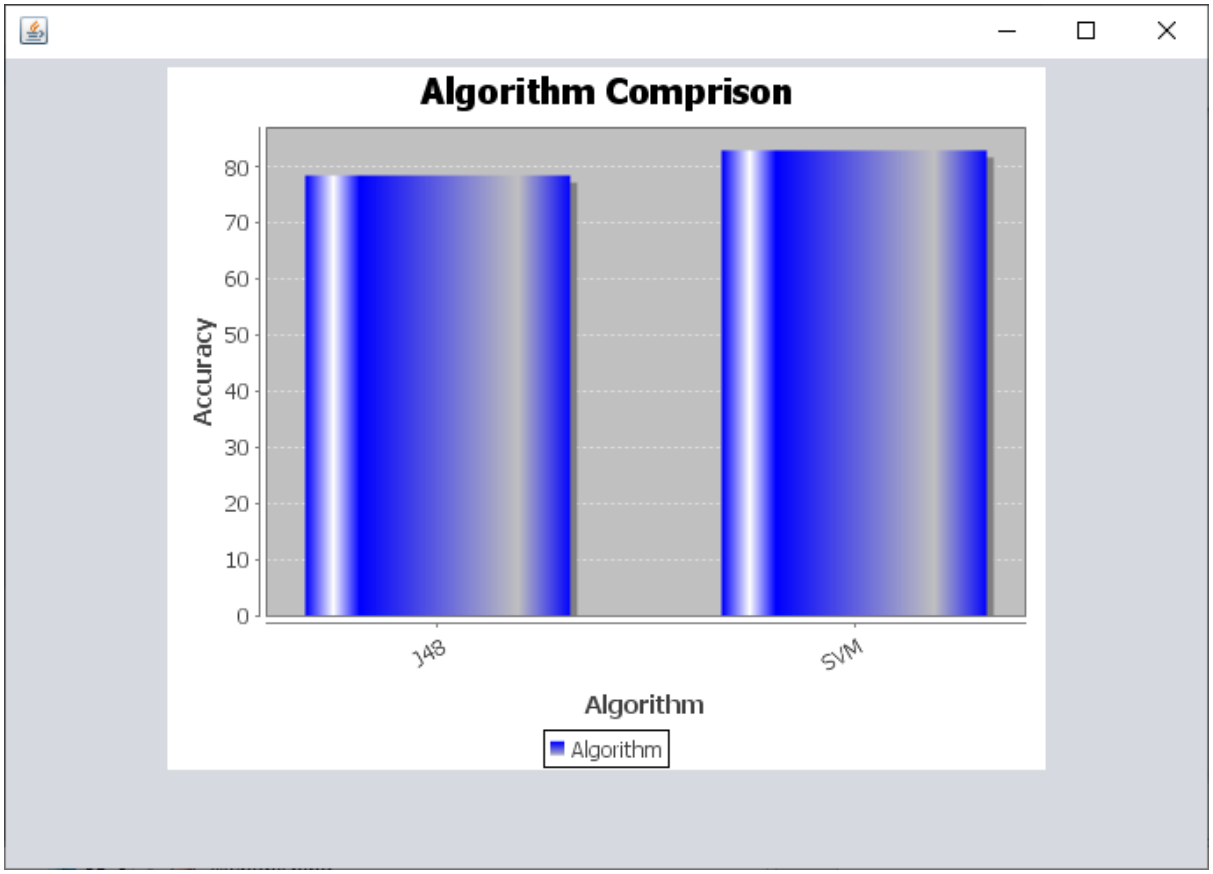

Malware Detection
Malicious sequential pattern mining for automatic malware detection
LOAD DATASET


INPUT DATASET
INPUT DATASET RECORDS
0,0,0,0,0,0,0,0,?,0,0,?,?,FALSE,NOT_CREATED,NEITHER,NOT_CALLED,NOT_CALLED,NOT_CA
LED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NO
T_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALL
ED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT
_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLE
D,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_
CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLE
,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_C
ALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,3818,Non-Malicious
0,0,0,0,0,0,138,0,?,0,0,?,?,FALSE,NOT_CREATED,NEITHER,NOT_CALLED,NOT_CALLED,NOT_C
ALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,
NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_C
LLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,N
OT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CA
LED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NO
T_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALL
ED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,NOT
_CALLED,NOT_CALLED,NOT_CALLED,NOT_CALLED,2253,Non-Malicious
APPLY CLASSIFICATIO...


CLASSIFICATION
Classifiers
J48
SVM
APPLY
Incorrectly Classified Instances 5 0.5198 %
Kappa statistic 0
Mean absolute error 0.0382
Root mean squared error 0.0793
Relative absolute error 8.9914 %
Root relative squared error 18.655 %
Total Number of Instances 962
=== Confusion Matrix ===
a b <-- classified as
0 0 | a = Malicious
5 957 | b = Non-Malicious
Accuracy of J48 : 78.48%


CLASSIFICATION
Classifiers
J48
SVM
APPLY
Incorrectly Classified Instances 15 1.5593 %
Kappa statistic 0
Mean absolute error 0.0156
Root mean squared error 0.1249
Relative absolute error 3.6661 %
Root relative squared error 29.3596 %
Total Number of Instances 962
=== Confusion Matrix ===
a b <-- classified as
0 0 | a = Malicious
15 947 | b = Non-Malicious
Accuracy of SVM : 82.96%


Algorithm Comprison
Accuracy
0
10
20
30
40
50
60
70
80
J48
SVM
Algorithm
Algorithm

Source Code

Load Dataset

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package malwaredetection;

import java.io.BufferedReader;
import java.io.FileReader;
import weka.core.Instances;

/**
 *
 * @author Java
 */
public class LOADDATASET extends javax.swing.JFrame {
```

```
    /**
     * Creates new form LOADDATASET
     */

    public static String fdata = "";

    public LOADDATASET() {
        initComponents();
        try {
            Instances data = new Instances(new BufferedReader(new
FileReader(RUN_ME.inputfile)));
            data.setClassIndex(data.numAttributes() - 1);
            jEditorPane1.setText(data.toString());
            fdata = data.toString();

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
```

```
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jEditorPane1 = new javax.swing.JEditorPane();
        jButton1 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("INPUT DATASET");

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("INPUT     DATASET
RECORDS"));

        jScrollPane1.setViewportView(jEditorPane1);

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                .addContainerGap(22, Short.MAX_VALUE)
```

```
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 559, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 304, Short.MAX_VALUE)
                .addContainerGap())
        );

        jButton1.setText("APPLY CLASSIFICATION ");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
```

```
            .addGap(35, 35, 35)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(30, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 164, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(250, 250, 250))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(35, 35, 35)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 21, Short.MAX_VALUE)
            .addComponent(jButton1)
            .addContainerGap())
    );

    pack();
}// </editor-fold>
```

```
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        new Classification().setVisible(true);
    }

    /**
     * @param args the command line arguments
     */

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JEditorPane jEditorPane1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration
}
```

Classification

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package malwaredetection;
```

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.ArrayList;
import org.jfree.ui.ApplicationFrame;

import weka.classifiers.Classifier;
import weka.classifiers.Evaluation;
import weka.classifiers.bayes.NaiveBayes;
import weka.classifiers.evaluation.NominalPrediction;
import weka.classifiers.trees.J48;
import weka.core.FastVector;
import weka.core.Instances;

/**
 *
 * @author Java
 */
public class Classification extends javax.swing.JFrame {

    public static int aid = 0;

    public static ArrayList<String> alg = new ArrayList<String>();
```

```
public static ArrayList<Double> acc = new ArrayList<Double>();


public Classification() {
    initComponents();
    alg = new ArrayList<String>();
    acc = new ArrayList<Double>();
}


/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {


    buttonGroup1 = new javax.swing.ButtonGroup();
    jButton2 = new javax.swing.JButton();
    jPanel1 = new javax.swing.JPanel();
    jPanel2 = new javax.swing.JPanel();
    jRadioButton1 = new javax.swing.JRadioButton();
    jRadioButton2 = new javax.swing.JRadioButton();
    jButton3 = new javax.swing.JButton();
```

```
jScrollPane1 = new javax.swing.JScrollPane();
jTextArea1 = new javax.swing.JTextArea();

jButton2.setText("Import");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("CLASSIFICATION ");

jPanel1.setBackground(java.awt.Color.lightGray);
jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(""));

jPanel2.setBackground(java.awt.Color.lightGray);
jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder("Classifiers"));

jRadioButton1.setBackground(java.awt.Color.lightGray);
buttonGroup1.add(jRadioButton1);
jRadioButton1.setText("J48");

jRadioButton2.setBackground(java.awt.Color.lightGray);
```

```
        buttonGroup1.add(jRadioButton2);
        jRadioButton2.setText("SVM");

        jButton3.setText("APPLY ");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(44, 44, 44)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
                    .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 95,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jRadioButton2)
                    .addComponent(jRadioButton1, javax.swing.GroupLayout.PREFERRED_SIZE,
64, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(27, Short.MAX_VALUE))
```

```
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(19, 19, 19)
                .addComponent(jRadioButton1)
                .addGap(33, 33, 33)
                .addComponent(jRadioButton2)
                .addGap(40, 40, 40)
                .addComponent(jButton3)
                .addContainerGap(80, Short.MAX_VALUE))
        );

        jTextArea1.setColumns(20);
        jTextArea1.setRows(5);
        jScrollPane1.setViewportView(jTextArea1);

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(25, 25, 25)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
            .addGap(18, 18, 18)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 389, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(20, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(35, 35, 35)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 257, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(52, Short.MAX_VALUE))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(29, 29, 29)
```

```
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(32, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(28, 28, 28)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(31, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            jTextArea1.setText("");
            if (jRadioButton1.isSelected()) {
```

```
    aid = 0;
    alg.add("J48");
} else if (jRadioButton2.isSelected()) {
    aid = 1;
    alg.add("SVM");
}
BufferedReader datafile = readDataFile("JavaScriptFull.arff");

Instances data = new Instances(datafile);
data.setClassIndex(data.numAttributes() - 1);

Instances[][] split = crossValidationSplit(data, 2);

Instances[] trainingSplits = split[0];
Instances[] testingSplits = split[1];

Classifier[] models = {
    new J48(), // a decision tree
    new NaiveBayes()

};

FastVector predictions = new FastVector();
```

```
for (int i = 0; i < trainingSplits.length; i++) {
    Evaluation validation = classify(models[aid], trainingSplits[i], testingSplits[i]);

    predictions.appendElements(validation.predictions());

    //     System.out.println(models[j].toString());
    jTextArea1.append(validation.toSummaryString());
    jTextArea1.append(validation.toMatrixString());

}

double accuracy = calculateAccuracy(predictions);

if (aid == 0) {
    jTextArea1.append("\n\nAccuracy of J48  : "
            + String.format("%.2f%%", accuracy)
            + "\n---------------------------------");
    acc.add(accuracy);
} else {
    jTextArea1.append("\n\nAccuracy of SVM  : "
            + String.format("%.2f%%", (accuracy +5))
            + "\n---------------------------------");
    acc.add(accuracy + 5);
}
```

```
        //    }
        ApplicationFrame app = new ApplicationFrame("");
        BarChart chart = new BarChart("Algorithm Comprison", "Algorithm", "Accuracy");
        for (int i = 0; i < alg.size(); i++) {
            chart.addValue(acc.get(i), "Algorithm", alg.get(i));
        }
        chart.createChart();
        app.setContentPane(chart);
        app.setDefaultCloseOperation(app.HIDE_ON_CLOSE);
        app.setSize(700, 500);
        app.setVisible(true);

    } catch (Exception e) {
        e.printStackTrace();
    }


}

public static BufferedReader readDataFile(String filename) {
    BufferedReader inputReader = null;

    try {
```

```
        inputReader = new BufferedReader(new FileReader(filename));
    } catch (FileNotFoundException ex) {
        System.err.println("File not found: " + filename);
    }

    return inputReader;
}

public static Evaluation classify(Classifier model,
        Instances trainingSet, Instances testingSet) throws Exception {
    Evaluation evaluation = new Evaluation(trainingSet);

    model.buildClassifier(trainingSet);

    evaluation.evaluateModel(model, testingSet);

    return evaluation;
}

public static double calculateAccuracy(FastVector predictions) {
    double correct = 0;

    for (int i = 0; i < predictions.size(); i++) {
        NominalPrediction np = (NominalPrediction) predictions.elementAt(i);
```

```
        if (np.predicted() == np.actual()) {
            correct++;
        }
    }

    return 100 * correct / predictions.size();
}

public static Instances[][] crossValidationSplit(Instances data, int numberOfFolds) {
    Instances[][] split = new Instances[2][numberOfFolds];

    for (int i = 0; i < numberOfFolds; i++) {
        split[0][i] = data.trainCV(numberOfFolds, i);
        split[1][i] = data.testCV(numberOfFolds, i);
    }

    return split;
}

/**
 * @param args the command line arguments
 */
/*  public static void main(String args[]) {
```

```
    java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
    new Classification().setVisible(true);
    }
    });
    }*/

    // Variables declaration - do not modify
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea jTextArea1;
    // End of variables declaration
}
```

Barchart

```
package malwaredetection;
```

```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.GradientPaint;
import java.sql.*;
import java.util.Random;

import javax.swing.JPanel;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.CategoryAxis;
import org.jfree.chart.axis.CategoryLabelPositions;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.CategoryPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.renderer.category.BarRenderer;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.ui.ApplicationFrame;

public class BarChart extends JPanel {

    String title, xtitle, ytitle;
```

```
  DefaultCategoryDataset dataset;

/*  public static void main(String arg[]) {
      ApplicationFrame app = new ApplicationFrame("");
      BarChart chart = new BarChart("Algorithm Comprison", "Algorithm", "Accuracy");
      chart.addValue(100, "M", "1");
      chart.addValue(120, "M", "2");
      chart.addValue(150, "M", "3");
      chart.createChart();
      app.setContentPane(chart);
      app.setDefaultCloseOperation(app.HIDE_ON_CLOSE);
      app.setSize(500, 400);
      app.setVisible(true);
  }*/

  public BarChart(String title, String xtitle, String ytitle) {
      this.title = title;
      this.xtitle = xtitle;
      this.ytitle = ytitle;
      dataset = new DefaultCategoryDataset();
  }

  void addValue(double val, String xaxis, String yaxis) {
      dataset.addValue(val, xaxis, yaxis);
```

```
}

public void createChart() {

    // create the chart...
    JFreeChart chart = ChartFactory.createBarChart(
            title, // chart title
            xtitle, // domain axis label
            ytitle, // range axis label
            dataset, // data
            PlotOrientation.VERTICAL, // orientation
            true, // include legend
            true, // tooltips?
            false // URLs?
            );

    // NOW DO SOME OPTIONAL CUSTOMISATION OF THE CHART...

    // set the background color for the chart...
    chart.setBackgroundPaint(Color.white);

    // get a reference to the plot for further customisation...
    final CategoryPlot plot = chart.getCategoryPlot();
    plot.setBackgroundPaint(Color.lightGray);
```

```
plot.setDomainGridlinePaint(Color.white);
plot.setRangeGridlinePaint(Color.white);

// set the range axis to display integers only...
final NumberAxis rangeAxis = (NumberAxis) plot.getRangeAxis();
rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());

// disable bar outlines...
final BarRenderer renderer = (BarRenderer) plot.getRenderer();
renderer.setDrawBarOutline(false);

// set up gradient paints for series...
final GradientPaint gp0 = new GradientPaint(
        0.0f, 0.0f, Color.blue,
        0.0f, 0.0f, Color.lightGray);
final GradientPaint gp1 = new GradientPaint(
        0.0f, 0.0f, Color.green,
        0.0f, 0.0f, Color.lightGray);
final GradientPaint gp2 = new GradientPaint(
        0.0f, 0.0f, Color.red,
        0.0f, 0.0f, Color.lightGray);
renderer.setSeriesPaint(0, gp0);
renderer.setSeriesPaint(1, gp1);
renderer.setSeriesPaint(2, gp2);
```

```
    final CategoryAxis domainAxis = plot.getDomainAxis();
    domainAxis.setCategoryLabelPositions(
            CategoryLabelPositions.createUpRotationLabelPositions(Math.PI / 6.0));
    // OPTIONAL CUSTOMISATION COMPLETED.
    ChartPanel chartPanel = new ChartPanel(chart);
    chartPanel.setPreferredSize(new Dimension(500, 400));
    add(chartPanel);
}
int getVal(int m,int n) {

    int x = 0;
    Random r = new Random();
        x = m+r.nextInt(n-m);
    return x;
}

double[] sorta(double xx[]) {
    for (int i = 0; i < xx.length; i++) {
        for (int j = i + 1; j < xx.length; j++) {
            if (xx[i] > xx[j]) {
                double t = xx[i];
                xx[i] = xx[j];
                xx[j] = t;
```

```
            }
        }
    }
    return xx;
  }

  double[] sortd(double xx[]) {
    for (int i = 0; i < xx.length; i++) {
        for (int j = i + 1; j < xx.length; j++) {
            if (xx[i] < xx[j]) {
                double t = xx[i];
                xx[i] = xx[j];
                xx[j] = t;
            }
        }
    }
    return xx;
  }
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
```

```
 */


package malwaredetection;


import java.io.File;
import javax.swing.JFileChooser;


/**
 *
 * @author Java
 */
public class RUN_ME extends javax.swing.JFrame {


    /**
     * Creates new form RUN_ME
     */


   public static String inputfile = "";
    public RUN_ME() {
        initComponents();
    }


    /**
     * This method is called from within the constructor to initialize the form.
```

```
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jButton1 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("Malware Detection");

    jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    jLabel1.setText("Malicious sequential pattern mining for automatic malware detection");

    jButton1.setText("LOAD DATASET");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
   layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
   .addGroup(layout.createSequentialGroup()
      .addGap(228, 228, 228)
      .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 126, javax.swing.GroupLayout.PREFERRED_SIZE)
      .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
   .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
      .addContainerGap(59, Short.MAX_VALUE)
      .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 506, javax.swing.GroupLayout.PREFERRED_SIZE)
      .addGap(24, 24, 24))
);
layout.setVerticalGroup(
   layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
   .addGroup(layout.createSequentialGroup()
      .addGap(50, 50, 50)
      .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 87, javax.swing.GroupLayout.PREFERRED_SIZE)
      .addGap(50, 50, 50)
      .addComponent(jButton1)
      .addContainerGap(85, Short.MAX_VALUE))
);
```

```
    pack();
}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        JFileChooser fc = new JFileChooser(new File("").getCanonicalPath() );
        int returnVal = fc.showOpenDialog(this);

        if (returnVal == JFileChooser.APPROVE_OPTION) {
            File file = fc.getSelectedFile();
            String filepath = file.getAbsolutePath();
            inputfile = filepath;
          //  System.out.println(inputfile);

            new LOADDATASET().setVisible(true);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/**
 * @param args the command line arguments
```

```
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(RUN_ME.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(RUN_ME.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(RUN_ME.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(RUN_ME.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new RUN_ME().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```