**流程图**

检测模块

UI

获取文件名

数据挖掘

构造命令

检测数据

调用检测模块

返回结果

显示检测结果

**具体函数流程**

**病毒检测模块：VirusDetection-2.0.py**

**# 从PE文件中提取数据值并将其作为成员变量保存的类**

**class PEFile:**

**#结果判断函数：**

**def printResult(classification):**

**#第一种检测方法**

**#J48()：C4.5决策树算法（决策树在分析各个属性时，是完全独立的） 及 数据判断**

**def runJ48():**

**#第二种weka数据挖掘树算法 及 数据判断**

**def runJ48Graft():**

**#第三种weka数据挖掘规则算法 及 数据判断**

**def runPART():**

**#第四种weka数据挖掘规则算法 及 数据判断**

**def runRidor():**

**#创建解析器对象ArgumentParser，可以添加参数**

**#add\_argument()方法，用来指定程序需要接受的命令参数**

**parser = argparse.ArgumentParser(description='Classify an unknown binary as MALWARE or CLEAN.')**

**#帮助函数**

**If not args.f:**

**parser.print\_help()**

**sys.exit(0)**

**input = PEFile(args.f)**

**# 实例化数据挖掘函数**

**if(args.v):**

**input.DataDump()**

**#帮助函数**

**if not args.n:**

**args.n = 0**

**args.n = int(args.n)**

**if args.n < 0 or args.n > 4:**

**parser.print\_help()**

**sys.exit(0)**

**# 判断输入的args并使用指定的算法**

**# Options 0: Run all models**

**if(args.n == 0):**

**# Options 1: Run J48**

**if(args.n == 1):**

**# Options 2: Run J48**

**if(args.n == 2):**

**# Option 3: Run PART**

**if(args.n == 3):**

**# Option 4: Run Ridor**

**if(args.n == 4):**

**UI模块：UI.py**

**#增加窗体属性**

**class filedialogdemo(QWidget):**

**#按钮类**

**self.myButton = QtWidgets.QPushButton(self)**

**#文本框类**

**self.content = QTextEdit()**

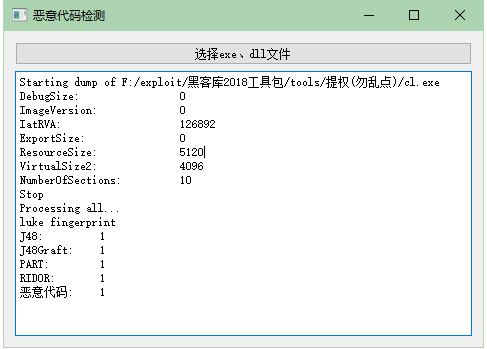
**#设置按钮调用函数**

**#功能：读取所选择文件的文件名，构造cmd命令调用恶意代码检测模块，收集返回信息并显示到文本框**

**def load\_text(self):**

**实例**

**含恶意代码EXE：**



**不含恶意代码EXE：**

