MULTI-AGENT SYSTEMS
Design Report

<u>TEAM: 04</u>
Anne Constanze
Joaquim Marset
Genevieve Masioni
Shivani Patel
Ruben Vera

Table of contents

1. Introduction

This project presents the idea of developing a Multi-Agent System that can classify if a particular firm is fraudulent or not, based on present and past risk factors. The agent-based decision support system (A-DSS) would be used by audit companies that want an automatic mechanism to detect fraudulent activities in a firm.

A Multi-Agent System (MAS) is defined by being a system that includes multiple agents that interact with each other to fulfill a certain task. An agent can be defined as a computer system that is situated in an environment and that is capable of taking actions and decisions by itself to fulfill its tasks. (Wooldridge, 2002, p.3)

In order to create the system, we will be using the UCI's Audit dataset, which contains different risk factors and tries to predict if a particular instance (a firm) is fraudulent or not.

We will also be using Weka, open-source software that provides different tools for data processing, machine learning algorithms and visualization tools (Tutorialspoint). This assignment will use its machine learning algorithms, as there is the necessity to train a certain number of classifiers that can perform the aforementioned prediction. In particular, each classifier will use the same J48 algorithm, an open-source Java class for generating a pruned or unpruned C4.5 decision tree (Frank). Once we have a prediction for each classifier, we will need a voting mechanism to produce the final prediction that the user will receive.

2. System Requirements

The requirement that needs to be considered for this project is the necessity of at least ten classifier agents, which will be trained using the WEKA algorithms, and performance metrics will be computed. Furthermore, the user agent, which will be in charge of requesting to classify the instances of the test set. Also, we can have other agents to facilitate the communication between agents or perform various tasks to have a well-balanced functioning system.

3. Design

General architecture

The overall architecture of the agent-based decision support system (A-DSS) is in Fig. 1. With this figure, an explanation of the architecture, in addition to the properties, function and sub-architecture of each agent type will be given.

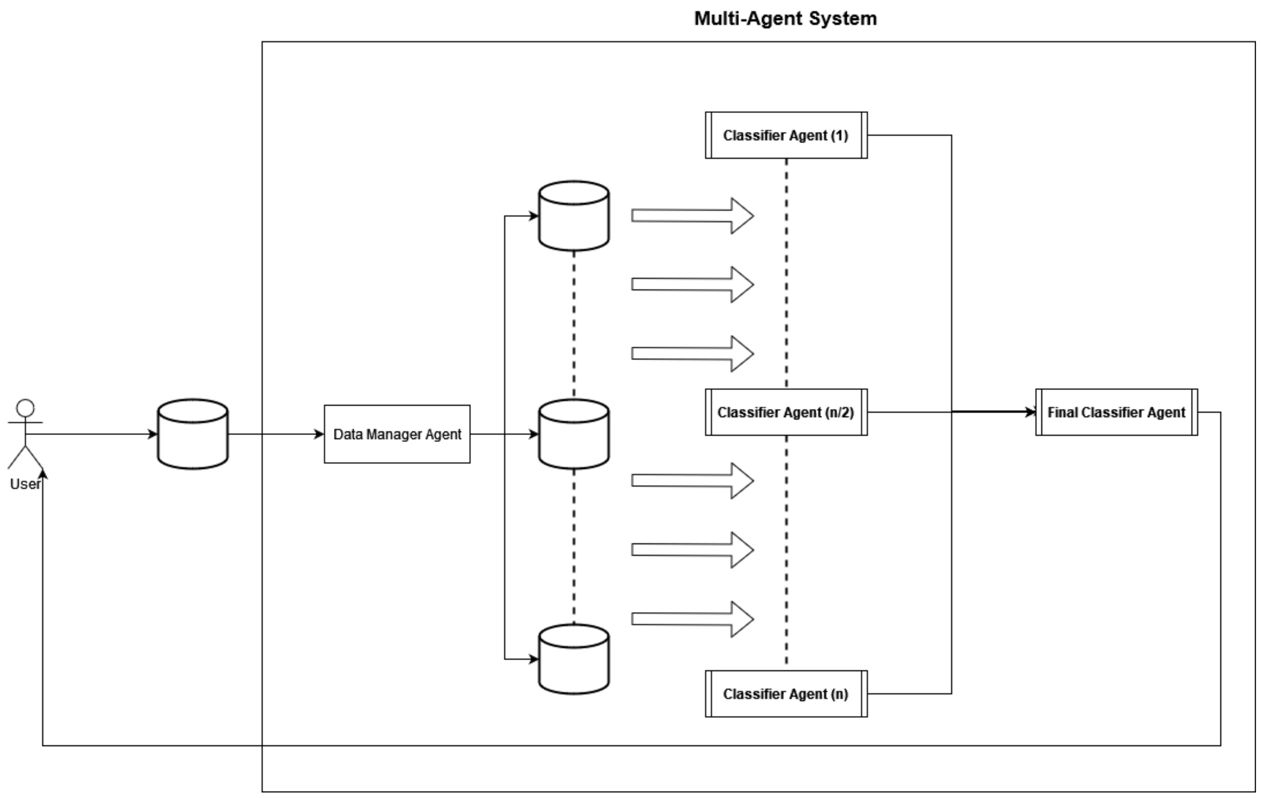

Fig. 1 Design of the Multi-Agent System

User Agent

As the name implies, this is the agent that will communicate with the user who wants to predict if a particular firm is fraudulent or not. It will be the only agent that will communicate with the user, and will be the starting and end point of the entire execution flow.

The user agent will be the one in charge of receiving the XML configuration files that the system will need in order to start, as well as the one used to perform the queries. The user will decide the configuration the system will have, as well as the different instances and attributes s/he wants to predict.

Once it has the configuration to initiate the system, it will inform the Data Divider Agent about the dataset file it should use to train the classifiers, the number of features each classifier should use, and the number of classifiers.

Then, once the system is ready to accept predictions, it will inform the Data Divider Agent about the instances we need to predict, delegating all the work and waiting for the final prediction.

Architecture

This agent will present a **reactive** architecture, given that we only want this agent to act when the user asks the system to perform a query. Moreover, it will also act when the final prediction for each instance comes from another agent, as explained further in the report. We do not need this agent to have long-term goals or create any kind of plan, as it mainly delegates work or performs simple tasks.

Properties

- *Reactive*: We are considering a static environment where we only have the agents that conform our system, and the human agent that wants to use the system. Given that this is the only agent that communicates with the human agent, it is the one from whom we must assure a continuous interaction with the environment, and a fast response time.

- *Social Ability*: This agent interacts with the human agent, the Data Divider Agent, and the Result Agent, hence this property. The interaction with the human agent might require a different language to the one used with the software agents.

- *Rational*: This agent will only act towards handling the user queries, and it will never act against it, nor will it give fake results.

- *Autonomous*: This agent will perform its tasks without user interaction or guidance, only restricted by the user configuration.

Type

We consider it an **Interface Agent**, given that this agent will be the one communicating with the user, receiving the query, as well as displaying the final prediction (i.e. getting the input and returning the output). It will neither learn nor be proactive, as it will not perform any recommendation or learn user preferences, but only serve queries.

Data Manager Agent

When the system starts, this agent will be responsible for receiving the training dataset made of 717 out of the 767 instances (the remaining 50 will be used for testing later on). Later, it will divide the dataset into subsets of 300 instances, and 6 attributes out of the initial 25, receiving each classifier one of these subsets.
Each time it divides the dataset into manageable sizes for the classifier agents, it will temporarily store the subsets until the task has been completed.
After the agent has finished dividing the provided dataset, it will pass the new subsets to various classifier agents.

Once the system has been trained, and the user performs a query with some instances to predict, this agent will decide which classifiers will receive each instance. The decision will depend on the subset of 20 features each test instance has, as well as the subset of 6 instances each classifier has been trained with. A classifier receives an instance to predict if the instance includes all the 6 features the classifier works with.

Architecture

The agent presents a **reactive** architecture, as it will perform a simple task that will not need any kind of planning. There is a small set of static rules that would be used for dividing and allocating the data into chunks of (300×6) for training, and the same for testing. However, the agent does not need to learn or demonstrate any kind of "intelligent" behavior to do so. The agent has a specific simple task that would not need any kind of abstract or complex rule structures.

Properties

- *Reactive*: The agent can receive different datasets given by the User Agent. The agent interacts with the said agent, and passes the processed data to the next agent, maintaining the interaction between them.
- *Social Ability*: The agent interacts with different agents within the system.
- *Rational*: Applies a set of rules while dividing the data and allocating them to different classifier agents. The agent will try to archive at any moment its task and will not act off of its delegated purpose within the system.
- *Autonomous*: This agent has autonomy as it does not need the user interaction to perform its designated tasks.

Type

This agent is a **Facilitator Agent**, and being more specific, it will act as a broker agent. This agent communicates the user agent with the classifier agents, and it organizes the size and number of attributes of the dataset.

Classifier Agent

This agent is the one that will embed the Weka J48 decision tree that will classify the different firm instances. We will have at least 10 instances of this agent, each one embedding this classification model. Each classifier will receive a training set of 300 instances and 6 attributes, leaving some of those instances as a validation set to assess their performance.

Each classifier will generate a set of performance metrics (e.g. accuracy) using the validation set that they will store, and later pass to the agent that will embed the coordination mechanism, as these metrics will serve to weigh the importance of each classifier in the final prediction.

Once the classifiers have been trained, the classifiers will receive test instances, if those instances have all the attributes the classifier needs, and will generate a prediction for each one. These predictions will be also handed to the coordination mechanism to generate the final prediction.

Architecture

This agent will follow a **hybrid** architecture, given that it will follow a plan to build the decision tree model (i.e. the Weka algorithm to create the decision tree). Nevertheless, once it has been trained it will act reactively, without complex reasoning, waiting for instances to classify, obtaining the predictions, and communicating the results to other agents.

Properties

- *Learning*: The agent learns a model that is able to distinguish at some extent normal and altered data from a firm. Selecting the best feature at each step, the cutting point for each feature, the post-pruning it does with the validation set, can result in different trees with different performances. However, once the model has been built, it will not learn more if it does not receive more data outside the initial dataset to refine the resulting trees.

- *Proactive*: As a classifier, these agents have a goal-directed behavior (computing predictions for a subset of data). While learning, it takes initiative, when necessary, to ensure the best classification performance.

- *Reasoning*: We can consider that this agent reasons, given that it uses its knowledge-base (i.e. the model) to infer the predictions for each queried instance.

- *Autonomous*: This agent is autonomous because it builds, and later traverses the model on its own to give predictions, without the help of the user.

- *Rational*: This agent is rational, as it will never give a prediction that is contrary to what its model says. It can happen that the model is not good enough and the prediction is incorrect. However, the agent is still rational.

- *Social Ability*: This agent needs to interact with the Data Divider as well as the Final Classifier.

Type

This agent is a **Wrapper Agent**, given that it is a decision tree model that is wrapped in JADE code. The agent is also able to interact with the other agents, receiving the datasets and features, and returning the predictions and performance measures.

Final Classifier Agent

This agent's purpose is to compute the ultimate prediction, which corresponds to the A-DSS' response for the user's data set. The final classifier agent receives as input predictions of previous classifiers (e.g. intermediate classifiers), the firms associated with those predictions, as well as the intermediate classifier's performance metrics (e.g. accuracy). The Final Classifier weights each prediction based on the corresponding intermediate classifier performance metrics and computes the ultimate prediction as a linear combination of previous predictions.

Being a classifier itself, the Final Classifier Agent learns to adjust the weights in order to automatically improve its performance and better its prediction.

Architecture

This agent has a **hybrid** architecture, given that it will first show a deliberative one to create the model that will weight each intermediate classifier based on their performance metrics. Then, once the model has been built, it will show a reactive behavior to return the final prediction of each received test instance.

Properties

- *Learning:* Given that it learns a set of weights for each intermediate classifier, it will try to learn those that give the best performance when obtaining the final prediction.
- *Proactive:* Has the goal of learning a model that best classifies instances, so it proactively tries to learn the model with the best performance.
- *Social ability*: Works in a coordinated way with all previous classifier agents, as well as with the result agent.
- *Reactive*: Maintains an ongoing interaction with its environment and responds to changes that occur.
- *Autonomous*: Has agency to make decisions, decide on which weight to give to the predictions of a specific agent, to achieve its goal efficiently, and the best possible performance metrics.

- *Rational:* It always gives a prediction using the learned model, so it will never give one contrary to its objective of classifying firm instances.
- *Social Ability:* This agent needs to interact with all the Classifier agents, as well as with the Result agent.

Type

This agent is a particularly collaborative **Wrapper Agent** because it is deliberative with reasoning capabilities that solves a classification problem too large for a single agent, with an emphasis on communication and cooperation with other classifier agents.

Agent table

Name	**Type**	**Function(s)**	**Properties**
User Agent	Interface Agent	- Interacts with the user - Receives the data set - Displays the system's prediction	Reactive, Social Ability, Rational, Autonomous
Data Divider Agent	Broker	- Divides the train data in subsets of 300 instances and 6 attributes - Creates new data sets with these subsets - Passes the new datasets to the classifiers - Passes the test instances to a classifier if the instance contains the classifier attributes	Reactive, Social Ability, Rational, Autonomous
Classifier Agents (>= 10)	Wrapper	- Builds a decision tree with the train data set - Stores performance metrics computed using a validation set - Classifies data into fraudulent or normal	Learning, Proactive, Reasoning, Rational, Autonomous, Social Ability

		- Outputs test predictions and performance metrics to the coordination mechanism	
Final Classifier Agent	Collaborative	- Builds a model/function that weights each classifier using their performance metrics - Receives previous classifier predictions for each instance - Receive firms that correspond to the predictions - Obtains a final prediction for each instance - Passes the final predictions to the Result agent	Learning, Proactive, Reactive, Reasoning, Rational, Autonomous, Social Ability
Result Agent	Information Agent	- Retrieves output information from final Classifier Agent - Sends the final prediction to the User Agent	Social Ability, Rational, Autonomous

References

- Ministry of Electronics and Information Technology (MEITY) (2018). https://archive.ics.uci.edu/ml/datasets/Audit+Data
- Frank, Eibe (NA). Weka J48 Documentation. WEKA Sourceforge. https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/J48.html
- Martín, Mario (2018). Data Mining 3: Decision Trees. https://www.cs.upc.edu/~mmartin/DM4%20-%20Decision%20trees.pdf
- TutorialPoint (NA). What is WEKA?. https://www.tutorialspoint.com/weka/what_is_weka.htm
- Wooldridge, MIchael (2002, April). Intelligent Agents: The Key Concepts. Springer, Berlin, Heidelberg. https://campusvirtual.urv.cat/pluginfile.php/3953480/mod_resource/content/1/wooldridge95intelligent_weiss.pdf
- Batet et al (2012). Turist@: Agent-based personalised recommendation of tourist activities.

https://campusvirtual.urv.cat/pluginfile.php/3953499/mod_resource/content/2/Turist%40%20ESWA%202012.pdf