Designing computer-assisted problem-based learning (CAPBL) environment for performance analysis of isolation forest algorithm

R D Hastuti, I Waspada, P W Wirawan, and N Bahtiar

Departemen of Computer Science, Faculty of Science and Mathematics, Diponegoro University
Jl. Prof. H. Soedarto, S.H., Tembalang, Semarang 50275, Indonesia

Corresponding author: rinadwihastuti27@gmail.com

Abstract. Anomaly detection is important applied in various fields of application to determine errors in the system. By detecting it can minimize losses on the system. Anomaly detection can be done using several algorithms. One such algorithm is the Isolation Forest algorithm. Isolation Forest Algorithm is an efficient and effective algorithm in detecting anomalies. Isolation Forest has better performance than other algorithms in terms of execution time, especially in large datasets. Although the Isolation Forest algorithm has many advantages, there are still very few tools that provide this algorithm. One tool that provides the Isolation Forest algorithm is Scikit Learn. However, using special tools such as Scikit Learning requires sufficient time and experience to be able to manage the features of the tool. Thus, in this study the authors developed a web-based application that is used to assist users in assisting and improving the performance of Isolation Forests by using sample data sets, setting the parameters of Isolation Forests, visualization, and evaluation. This application was developed using the waterfall process model and CAPBL concept. In determining the features of the application, an analysis of features is based on the CAPBL concept with the business process of CRISP-DM. The software testing result shows that this application is fit to be used as a new solution to facilitate users who want to learn and analyze the performance of Isolation Forest.

1. Introduction
Anomaly is a pattern of data that has different characteristics from normal data [1]. To recognize anomalies in data it is necessary to identify data patterns using anomaly detection techniques in data mining. Anomaly detection is important applied in various fields of application to determine errors in a system. By knowing these anomalies can minimize these errors. Algorithms that can be used to detect anomalies include ORCA, Local Outlier Factor (LOF), One-Class SVM (OCSVM), Random Forest (RF), and Isolation Forest. From some examples of these algorithms, one of the anomaly detection algorithms that is taken into account today is Isolation Forest [2].

The Isolation Forest algorithm is efficient and effective in handling anomaly detection. This algorithm is based on a binary tree structure that builds a series of isolation trees (iTree) for data that is determined through random sampling[3]. Isolation Forest has better performance than other algorithms in terms of execution time, especially in large datasets. In research conducted by Domingues [2], prove that Isolation Forest is superior in terms of ROC, precision-recall, robustness, complexity, and memory usage compared to other anomaly detection algorithms.
Although the Isolation Forest algorithm has many advantages, there are still a few tools that provide the algorithm. One of the tools that provide the Isolation Forest algorithm is Scikit Learn. However, the use of certain tools such as Scikit Learn requires sufficient time and experience to be able to master the features of these tools[4]. This makes it difficult for users who do not master the programming language to use the Isolation Forest algorithm.

Thus, in this study the development of an Isolation Forest performance analysis application as an anomaly detection is used to assist users in learning and observing the performance of the Isolation Forest algorithm with accompanying dataset examples, setting Isolation Forest parameters, visualization and evaluation results.

Application of performance analysis of Isolation Forest as an anomaly detection was developed with the concept of Computer Assisted Problem Based Learning (CAPBL) based on business processes that refer to the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology. The main purpose of using CAPBL is to provide practicality of assistance to users in the learning process. According to Rajeswari[5], The CAPBL model has proven to be effective in improving students' thinking skills in developing three components of expertise, namely critical, creative thinking, and problem-solving skills. However, to implement CAPBL requires an understanding of business processes in detecting anomalies using Isolation Forest. Meanwhile, Isolation Forest itself is one of the algorithms in the study of anomaly detection in the field of data mining. So in this study, the CRISP-DM methodology is used as a business process to guide the stages of structured data mining.

2. Terms in the study

2.1. Isolation forest
Isolation forest, or iForest, is an anomaly detection algorithm that is based on the assumption that anomaly data is scarce and has a large data distance from normal data. The forest algorithm is efficient and effective in handling anomaly detection based on binary tree structures and building a series of trees for data that is determined through random sampling [3].

2.2. Library isolation forest pada scikit learn
The Isolation Forest Library in the Scikit Learn tool contains several important parameters. The description of the Isolation Forest library parameters is resumed in Table 1.

| Parameter          | Description                                                                 | Default Value |
|--------------------|------------------------------------------------------------------------------|---------------|
| n_estimators       | The number of base estimators in the ensemble                                | 100           |
| max_samples        | The number of samples to draw from X to train each base estimator. If max_samples are larger than the number of samples provided, all samples will be used for all trees (no sampling). | 256           |
| Contamination      | The amount of contamination of the data set, i.e. the proportion of outliers in the data set. Used when fitting to define the threshold on the decision function. If ‘auto’, the decision function threshold is determined as in the original paper. | 0.1           |
| max_features       | The number of features to draw from X to train each base estimator. If True, individual trees are fit on random subsets of the training data sampled with replacement. If False, sampling without replacement is performed. | 1.0           |
| Bootstrap          | The number of jobs to run in parallel for both fits and predict. None means 1 unless in a joblib.parallel_backend context. -1 means using all processors. | None          |
### Parameter Description

| Parameter | Description                                                                                                                                                                                                 | Default Value |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| behavior  | The behavior of the decision_function which can be either ‘old’ or ‘new’. Passing behaviour=’new’ makes the decision_function change to match other anomaly detection algorithm API which will be the default behavior in the future. As explained in detail in the offset_attribute documentation, the decision_function becomes dependent on the contamination parameter, in such a way that 0 becomes its natural threshold to detect outliers. | ‘old’         |
| random_state | If int, random_state is the seed used by the random number generator; If RandomState instance, random_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random. | None           |
| verbose   | Controls the verbosity of the tree building process.                                                                                                                                                          | 0             |
| warm_start | When set to True, reuse the solution of the previous call to fit and add more estimators to the ensemble, otherwise, just fit a whole new forest.                                                               | False         |

In addition to the parameters mentioned, the Isolation Forest library has several methods used in the modeling process. These methods include [6]:

1. *fit*, as an estimator to train data in dataset X using the Isolation Forest algorithm.
2. *predict*, predict certain samples whether included outliers or not.
3. *fit_predict*, run the fit command on dataset X and then return the label on dataset X.

#### 2.3. Problem-based learning (PBL)

Problem-based learning (PBL) is learning centered on the learner where the learner learns about a subject through problem-solving experience. The purpose of PBL is to help learners develop knowledge flexibly, develop effective problem-solving skills, learn independently, develop collaborative skills and intrinsic motivation[5].

#### 2.4. Cross-industry standard prosses for data mining (CRISP-DM)

Cross-Industry Standard Process for Data Mining (CRISP-DM) is a standardization of the process of data mining as a general problem-solving strategy of a business or research unit. The CRISP-DM life cycle is divided into six phases. A general CRISP-DM life cycle overview will be displayed in Figure 1.

![Figure 1. CRISP-DM methodology](image)
As illustrated in Figure 1, the life cycle of CRISP-DM consists of six phases, i.e.[7]:

1. **Business Understanding Phase**
   - Determine the project objectives and needs in detail in the overall scope of the business or research unit. Then translate the goals and constraints into formulas and definitions of data mining problems, followed by preparing an initial strategy for achieving the objectives.

2. **Data Understanding Phase**
   - Collecting data using data analysis investigations to identify more about the data and search for the initial knowledge in it. Then, evaluate the quality of the data.

3. **Data Preparation Phase**
   - Prepare data available from the start. Then choose the cases and variables that want to be analyzed and those that are by the analysis to be performed. After the case is selected, proceed with making changes to some variables if needed.

4. **Modeling Phase**
   - Select and apply appropriate modeling techniques. After that, the model rules are calibrated to optimize the results. If necessary, the process can return to the data processing phase to make the data into a form by the requirements specifications of certain data mining techniques.

5. **Evaluation Phase**
   - Evaluate one or more models used in the modeling phase to get quality and effectiveness before they are deployed for use. Then, determine whether there is a model that meets the objectives in the initial phase. Determine whether there are important issues from business or research that are not handled properly. Making decisions relating to the use of results from data mining.

6. **Deployment Phase**
   - Using the resulting model. The formation of the model does not indicate the completion of the project.

2.5. **Principal component analysis (PCA)**
Principal component analysis (PCA) can be described as a way of identifying patterns in data and expressing data in such a way as to show similarities and differences in data. PCA is a powerful tool for analyzing patterns in high-dimensional data. The main advantage of PCA is that it can compress data, which means reducing the number of dimensions without losing information. This approach has been recognized as a useful technique in data reduction, modeling, outlier detection, variable selection, classification, and prediction[8].

   Simply put, PCA is a common name for a technique that uses basic mathematical principles to change some variables that might correlate to a small number of variables called Principal Components (PCs)[9].

3. **Methods**
The research methodology carried out in this thesis can be seen in Figure 2. The following are detailed explanations for each part of the methodology.

3.1. **CAPBL Analysis**
The CAPBL analysis phase involves gathering requirements by applying the CAPBL concept to ensure that the application being built meets the needs as a CAPBL based application. To get the CAPBL-based application requirements analysis was carried out concerning the research design proposed by Qomarudin[10].

   However, to implement CAPBL requires an understanding of business processes in detecting anomalies using Isolation Forest. Isolation Forest itself is one of the algorithms in the study of anomaly detection in the field of data mining. So in this study, the CRISP-DM methodology is used as a business process to guide the stages of structured data mining.
3.2. Requirement analysis and definition
At this stage, the CRISP-DM methodology is implemented, so that a classification model with standard data mining process will be obtained later. The flow generated from CRISP-DM describes the business process in detecting anomalies with Isolation Forest. The business processes that have been described through CRISP-DM provide accurate information about activities at each stage. From the analysis of the CRISP-DM business process, the CAPBL concept can be applied at each stage so that the functional requirements of the software in the form of a System Requirement Specification (SRS) can be obtained as an application of CAPBL concept based Isolation Forest performance analysis.

3.3. System and software design.
System and software design is carried out after the requirements analysis and definition stages. At this stage includes data modeling, functional modeling, and interface design.

3.4. Implementation and unit testing.
This stage is carried out after the requirements analysis and definition and system and software design stages are completed. At this stage describes the implementation environment, implementation interface, and testing of applications have been developed.

4. Result and discussion

4.1. Application development
At the application development stage, it explains the results of CAPBL Analysis, Requirement Analysis and Definition, interface design, and implementation results of the application. Next is the explanation.

4.1.1. Result of CAPBL requirement analysis. Analysis that has been carried out at this stage produces three CAPBL requirements, namely independent learning, learning by providing problems, and interactive learning. These three requirements were chosen based on the CAPBL requirements which were most emphasized in the Qomaruddin [10]and Rajeswari [5]paper. These three requirements are presented in Table 2.

4.1.2. Result of CRISP-DM analysis. The results of the analysis using the CRISP-DM methodology for each phase are described as follows:
1. Business understanding. The business objective of this research is a new solution to facilitate users who want to use, study and observe the performance of Isolation Forest with an application. The
application is an Isolation Forest performance analysis application as an anomaly detection that is packaged with the CAPBL concept based on the CRISP-DM business process.

2. **Data understanding.** In this study data collection was done by downloading anomalous datasets. The reference dataset used in this study was taken from the sample dataset used in Liu's [1] and Domingues's [2] study. The dataset includes Smtp, Anthyroid, Satellite, Pima, and Cardio which can be accessed on the UCI Machine Learning Repository, Kaggle webpage and the Outlier Detection Dataset (ODDS) webpage [1], [2]. However, in this study, the selection of a dataset is used to identify the features of the application. The dataset used has characteristic attributes in the form of numbers and with class attributes that have been mapped to 0 if the class is a normal class and 1 if the class is an anomaly class and there is no limit to the number of attributes in the dataset used. Also, the dataset used is data that is ready for processing or data that has been carried out by the cleaning process so that there are no attributes that have values with missing values, null, or NaN. From these criteria, one dataset can be identified in this study, the Pima (Diabetes) dataset. This dataset consists of 768 lines and 9 attributes. From 768 rows there are 268 data samples with normal classes and there are 500 data samples with anomaly classes. For more details, the distribution of normal class values and anomalies are visualized using Principal Component Analysis (PCA) by reducing the diabetes dataset attribute to 2 attributes.

3. **Data preprocessing.** As already explained in the data understanding phase, that the data used is data that is ready though so there is no need for the data preprocessing phase in this application.

4. **Modeling.** In this case, the algorithm used is the Isolation Forest algorithm. The implementation of anomaly detection in this application uses the Isolation Forest library on the Scikit Learn tool. In the Isolation Forest library, two modeling methods can be done to get prediction results, which are using fit_predict or doing a fit process and then proceed with the predicted process (fit and predict). After the prediction process is done, the prediction results are obtained. Then the predicted data is compared with the original data visualized into the confusion matrix.

5. **Evaluation.** Evaluation of anomaly detection using the Isolation Forest algorithm is shown by the value of the confusion matrix that has been generated through the modeling phase. From the resulting confusion matrix, it is known the quality and performance possessed by the Isolation Forest algorithm by measuring the value of precision, recall and f1 score [1],[2].

Based on the analysis that has been done by implementing the CRISP-DM methodology, the application requirements needed in the application of CAPBL-based Isolation Forest algorithm performance analysis. The application requirements are presented in table 3.

**Table 2. Results of CAPBL requirement analysis**

| No | ID CAPBL      | CAPBL Requirements       | Application Requirements                                      |
|----|---------------|--------------------------|---------------------------------------------------------------|
| 1  | ID-CAPBL-01   | Independent Learning     | – The application displays general information of the Isolation Forest algorithm
|    |               |                          | – The application displays how to use each of the features available |
| 2  | ID-CAPBL-02   | Learning by providing problems | – The application can upload a dataset
|    |               |                          | – The application provides a sample dataset                  |
| 3  | ID-CAPBL-03   | Interactive learning     | – The application can display the results of the exploration of the dataset
|    |               |                          | – The application can display PCA
|    |               |                          | – The application provides modeling parameter settings        |
|    |               |                          | – The application displays an evaluation of the modeling      |
Table 3. Features results identification of CRISP-DM requirements

| No | ID CRISP-DM | CRISP-DM Stage         | Application Requirements                                                                 |
|----|-------------|------------------------|------------------------------------------------------------------------------------------|
| 1  | ID-CD-01    | Data understanding     | The application can upload a dataset                                                        |
|    |             | phase                  | The application can display data tables                                                     |
|    |             |                        | The application can display statistical data                                               |
|    |             |                        | The application can display PCA                                                            |
| 2  | ID-CD-02    | Modeling phase         | The application provides parameter settings for the Isolation Forest algorithm             |
|    |             |                        | The application can display Confusion Matrix                                               |
| 3  | ID-CD-03    | Evaluation phase       | The application can display evaluation metrics in the form of precision, recall, and f1 score. |

4.1.3. Functional requirement. From the results of application requirements that have been obtained at the CAPBL analysis stage and requirements analysis at the CRISP-DM stage, a Software Requirement Specification (SRS) can be prepared. Functional requirements in the form of SRS are shown in Table 4.

Table 4. Functional needs of isolation forest performance analysis application based on the CAPBL concept

| No | ID CAPBL | ID CRISP-DM | ID SRS      | Description                                                                                          |
|----|----------|-------------|-------------|-------------------------------------------------------------------------------------------------------|
| 1  | CAPBL-01 | -           | SRSIF-F-01  | Displays general application information                                                              |
| 2  | CAPBL-02 | CD-01       | SRSIF-F-02  | Uploading the dataset used in the experiment                                                           |
| 3  | CAPBL-03 | CD-01       | SRSIF-F-03  | Displays dataset information                                                                          |
| 4  | CAPBL-03 | CD-01       | SRSIF-F-04  | Showing statistical data                                                                               |
| 5  | CAPBL-03 | CD-01       | SRSIF-F-05  | Showing PCA                                                                                            |
|    |          |             |             | Determine the Isolation Forest algorithm parameter values (n_estimators, max_samples, contamination) used to predict data classes |
| 6  | CAPBL-03 | CD-02       | SRSIF-F-06  | Showing Confusion Matrix                                                                               |
| 7  | CAPBL-03 | CD-03       | SRSIF-F-07  | Displays the results of evaluation metrics in the form of precision, recall, and f1 score              |
| 8  | CAPBL-03 | CD-03       | SRSIF-F-08  | Showing application usage guidelines                                                                    |
| 9  | CAPBL-01 | -           | SRSIF-F-09  |                                                                                                       |

4.2. Examples of application use
To demonstrate the capabilities of the software that was built, in this section an experimental sample is given using the PIMA dataset (diabetes). The selection of dataset is based on the dataset reference contained in Liu's research[1]. This experiment carried out in several stages based on the order of features available on the application. The stages are presented in Figure 3.
Figure 3. Feature sequence of Pima dataset experiment in diabetes on the application of performance analysis of CAPBL based concepts

4.3. Experiment on pima dataset.

The system displays information in the form of data tables and statistical data. From the statistical data displayed, it can be seen the number of instances (data) in the table, the average, standard deviations, minimum values, 25%, 50% 75%, maximum values from the column dataset. From new attributes and contents of the diabetes dataset. Through Figure 5, can be known the amount of data for each column in the diabetes dataset consisting of 768 instances, with averages, standard deviations, minimum and maximum values that have been shown. On the data information page, the application can display anomalous and normal data distribution in the form of a two-dimensional PCA. Normal data on PCA is displayed in green while anomalous data is displayed in red. Data Display PCA information can be seen in Figure 6. The evaluation results in the form of a confusion matrix in trials using the Diabetes dataset are shown in Figure 7. From the results of the confusion matrix, the evaluation metric values are obtained.

The calculation of the value of the evaluation metric to find the best parameters is done in three stages, the first comparing the results of the evaluation metrics in different Number of Tree parameter settings. After finding the best results, the second stage compares the results of evaluation metrics on the Sample Size parameter. After getting the best value from the Number of Tree and Sample Size parameters, then the results of the evaluation metrics are compared to the contamination parameter. From the results of the comparison, the best value of the combination of the three parameters is shown with the highest f1 score. The parameter comparison results in the Isolation Forest trial for the diabetes dataset are shown in Table 5 to Table 7. Experiments with the best value on the parameter setting number of the tree that occurs when the value of the number of trees is 800. Experiments with the best value on setting the sample size parameter occur when the sample size value is 256. And for the experiment setting the contamination parameter obtained the best value when the contamination is at the value of 0.501. The results of the evaluation metrics from the combination of the three parameters produce a precision value of 0.679, recall 0.509, and f1 score of 0.582.
Figure 4. display of diabetes dataset table

Figure 5. Display diabetes diabetes statistical information
**Figure 6.** 2-Dimensional PCA display diabetes dataset

**Figure 7.** Display evaluation results with n_estimater 100, contamination 0.341, max_sample 256, and test size 0.3
Table 5. Comparison of number of tree parameters in the isolation forest experiment using diabetes dataset with the fit and predict methods.

| Number Of Tree | Sample Size | Contamination | Precision | Recall | F1 Score |
|----------------|-------------|---------------|-----------|--------|----------|
| 650            | 256         | 0.341         | 0.444     | 0.571  | 0.5      |
| 700            | 256         | 0.341         | 0.457     | 0.578  | 0.51     |
| 750            | 256         | 0.341         | 0.469     | 0.585  | 0.521    |
| 800            | 256         | 0.341         | 0.481     | 0.591  | 0.531    |
| 850            | 256         | 0.341         | 0.481     | 0.582  | 0.527    |
| 900            | 256         | 0.341         | 0.469     | 0.576  | 0.517    |
| 950            | 256         | 0.341         | 0.481     | 0.582  | 0.527    |
| 1000           | 256         | 0.341         | 0.481     | 0.574  | 0.523    |

Table 6. Results of comparison of sample size parameters in the isolation forest experiment using diabetes dataset with the fit and predict methods.

| Number Of Tree | Sample Size | Contamination | Precision | Recall | F1 Score |
|----------------|-------------|---------------|-----------|--------|----------|
| 80             | 64          | 0.341         | 0.42      | 0.586  | 0.489    |
| 800            | 128         | 0.341         | 0.444     | 0.571  | 0.5      |
| 800            | 256         | 0.341         | 0.481     | 0.591  | 0.531    |
| 800            | 512         | 0.341         | 0.481     | 0.549  | 0.513    |
| 800            | 1024        | 0.341         | 0.494     | 0.563  | 0.526    |
| 800            | 2048        | 0.341         | 0.494     | 0.563  | 0.526    |

Table 7. Comparison of contamination parameters in isolation forest experiments using diabetes dataset with the fit and predict methods.

| Number Of Tree | Sample Size | Contamination | Precision | Recall | F1 Score |
|----------------|-------------|---------------|-----------|--------|----------|
| 80             | 256         | 0.421         | 0.531     | 0.5    | 0.515    |
| 800            | 256         | 0.441         | 0.593     | 0.522  | 0.555    |
| 800            | 256         | 0.461         | 0.564     | 0.51   | 0.564    |
| 800            | 256         | 0.481         | 0.654     | 0.515  | 0.576    |
| 800            | 256         | 0.501         | 0.679     | 0.509  | 0.582    |
| 800            | 256         | 0.511         | 0.679     | 0.505  | 0.579    |
| 800            | 256         | 0.521         | 0.679     | 0.5    | 0.576    |

5. Conclusion

Based on the results of application development that has been carried out in the thesis of the Isolation Forest Performance Analysis Application Based on the CAPBL Concept it can be concluded that isolation Forest Performance Analysis application software has been developed with the identification of functional requirements through the CAPBL concept and CRISP-DM process analysis. This application can be used to facilitate and help understand the performance of the Isolation Forest algorithm in the various case examples provided. Observations can also be made through several integrated facilities, including statistical information from the dataset used, 2-dimensional PCA visualization, hyperparameter settings on the Isolation Forest algorithm, and also equipped with a visualization confusion matrix table and evaluation metric values in the form of precision, recall, and f1 score.

References
[1] Liu F T, Ting K M and Zhou Z-H 2012 ACM Trans. Knowl. Discov. Data vol. 6 no. 1 pp. 1–39
[2] Domingues R, Filippone M, Michiardi P and Zouaoui J 2018 Pattern Recognit. vol. 74 pp. 406–
[3] Ding Z and Fei M 2013 IFAC Proceedings Volumes (IFAC-PapersOnline) vol. 3 no. PART 1 pp. 12-17

[4] Özmen B and Altun A 2014 Turkish Online J. Qual. Inq. vol. 5 no. 3 pp. 9-27

[5] Rajeswari K 2015 J. Chem. Chem. Eng. vol. 9 pp. 233–237

[6] Scikit-learn developers, “sklearn.ensemble.IsolationForest.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html. [Accessed: 05-Jul-2019].

[7] Larose D T and Larose C D 2014 Discovering Knowledge in Data

[8] Ruby-Figueroa R 2015 Encyclopedia of Membranes

[9] Wang Q, Gao Q, Gao X and Nie F 2017 IJCAI Int. Jt. Conf. Artif. Intell. no. January pp. 2936–2942

[10] Qomaruddin M, Rahman A A and Iahad N A 2014 International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)