Covenant Journal of Informatics & Communication Technology. Vol. 8 No. 2, December., 2020

ISSN: *print* 2354 – 3566 *electronics* 2354 – 3507 DOI: An

Open Access Journal Available Online

Development of an Explainable Artificial Intelligence Prototype for Interpreting Predictive Models

Nnaemeka Udenwagu[1], Ambrose Azeta[1], Vivian Nwaocha[2], Victor Azeta[4], Daniel Enosegbe[3], Awal ganiyu[1], Adejoke Ajibade[1]

[1]Covenant University Ota
[2]National Open University Lagos
[3]Babcock University Ogun State
[4]National Productivity Center, Kaduna

nudenwagu@gmail.com, ambrose.azeta@covenantuniversity.edu.ng, onwaocha@noun.edu.ng, enosegbedan@gmail.com
awalgbere@gmail.com, adejokeajibade01@gmail.com, victaazeta@gmail.com

Received: 13.10.2020 Accepted: 30.11.2020
Date of Publication: December 2020

***Abstract*-**Artificial Intelligence (AI) now depends on black box machine learning (ML) models which lack algorithmic transparency. Some governments are responding to this through legislation like the "Right of Explanation" rule in the EU and "Algorithmic Accountability Act" in the USA in 2019. The attempt to open up the black box and introduce some level of interpretation has given rise to what is today known as Explainable Artificial Intelligence (XAI). The objective of this paper is to provide a design and implementation of an Explainable Artificial Intelligence Prototype (ExplainEx) that interprets predictive models by explaining their confusion matrix, component classes and classification accuracy. This study is limited to four ML algorithms including J48, Random Tree, RepTree and FURIA. At the core of the software is an engine automating a seamless interaction between Expliclas Web API and the trained datasets, to provide natural language explanation. The prototype is both a stand-alone and client-server based system capable of providing global explanations for any model built on any of the four ML algorithms. It supports multiple concurrent users in a client-server environment and can apply all four algorithms concurrently on a single dataset and returning both precision score and explanation. It is a ready tool for researchers who have datasets and classifiers prepared for explanation. This work bridges the

gap between prediction and explanation, thereby allowing researchers to concentrate on data analysis and building state-of-the-art predictive models.

.

Keywords/Index Terms: **Explainable, artificial intelligence, interpretable, machine learning, predictive models**

1. INTRODUCTION

For decades now Artificial Intelligence (AI) has been applied to Disease diagnosis (Stoel, 2019), Medical imaging (Alexander, et. al., 2019) , Cancer treatment (Ibrahim, et. al., 2020), Governance (Sharma, et. al., 2020), Geosciences (Jiao & Alavi, 2019), Economy (Constantinescu, et. al., 2019), Education (Amamou & Cheniti-belcadhi, 2018), Jurisprudence (Malgieri, 2019), Transportation and logistics (Siems-anderson, et. al., 2019), and Crime (Falade, et. al., 2019) and so on, in order to enhance productivity. In recent years AI has moved towards machine learning, which involves forecasting future results based on available data. Under varying circumstances, different machine learning (ML) algorithms have been known to provide certain levels of predictive accuracy. This has been used extensively in different domains, including prediction of crop yield and animal management in agriculture, diagnostic prediction in medicine, predicting passenger behavior in transportation industry, predicting criminal activities in safety and security. Prediction models are mostly built on machine learning (ML) algorithms such as Decision Trees, Artificial Neural Networks (ANN), Naïve Bayes, Linear regression and so on, most of which lack interpretability and algorithmic transparency and thus viewed as "black boxes".

When AI is applied to trivial use like gaming, humans have been known to trust its artificial expertise to a large extent. However, in critical sectors such as medicine and transportation, people have been reluctant to trust AI as much as human experts. Hence the increasing need for AI to acquire the capacity to explain its automated results to humans. In addition to this, some countries are beginning demand for explanations from AI results and predictions (Eoin, et. al., 2019). Thus this limitation has created a huge gap in the use of AI and hence the need for explainable AI (Dymitruk, 2019).Explainable AI is an attempt to introduce trust and transparency into AI to improve understandability in domains such as medicine, robotic engineering, education and adaptive learning, transportation and so on. This can be achieved through several methods, for example the Case-Based Reasoning (CBR) model in. Others have used the twin-based hybrid methods. Another method is to collect and infuse "Domain Knowledge" into a "black box" model like ANN to make it more interpretable. Yet other methods involves the use of open source frameworks like ExpliClas, Skater and so on, to interpret prediction results from decision trees and other machine learning algorithms (Lamy, et. al., 2019; Calvaresi & Framling, 2019; Eberle & Bundy, 2019).

2. REQUIREMENT MODELING

The Universal Modeling Language (UML) is used mainly to depict the interaction between a user and system components. The UML is a standard symbolic language that is used to represent a systems design and the interaction among its components.

2.1. Use Case Model

Figure 1 represents the actual use case for typical user of the system. The user first logs in into the system to start a session. Every session is uniquely identified by a session identification number. A user could start several sessions to handle different projects. The user goes on to upload a dataset and ML algorithm. Both the dataset and algorithm form part of the predictive model. The user finally builds a classifier and generates an explanation based on the uploaded model.

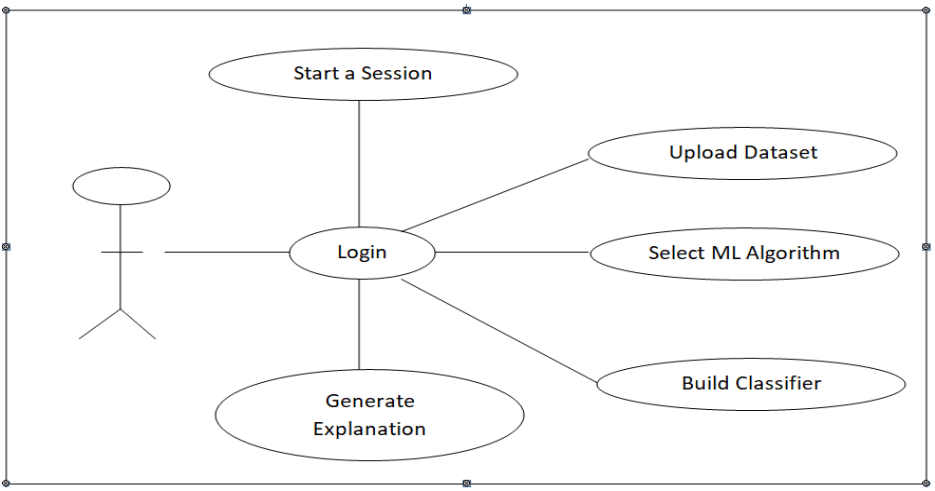

FIGURE 1. USE CASE DIAGRAM FOR USER

2.2. Class Structure Model

Figure 2 shows the class diagram of the system database. There are three components that make up each class namely, the class name, attributes and methods (the operations or functions carried out by each class). The association between the classes is depicted by the lines connecting the classes. Each user can start several independent sessions. Each session can only maintain a single project at a time. Each project returns only one set of result and each user can have several results stored in the database.

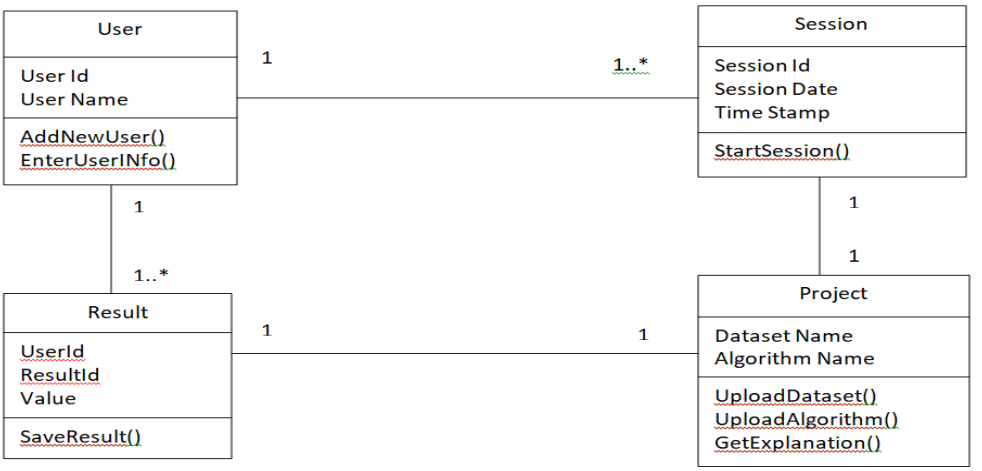

FIGURE 2. A CLASS DIAGRAM FOR EXPLAINEX DATABASE

2.3. *Sequence Model*

The sequence diagram in Figure 3 describes the process flow through the system's modules towards producing an explanation for predictive model provided by a user. The user begins the process by uploading a dataset and its associated ML algorithm through a web interface. The web interface passes the dataset and algorithm pair to the ExpliClas engine (API). The API in turn carries out the model interpretation and calls the natural language translator to provide a textual explanation. The final explanation is then passed back to the user following the same route.

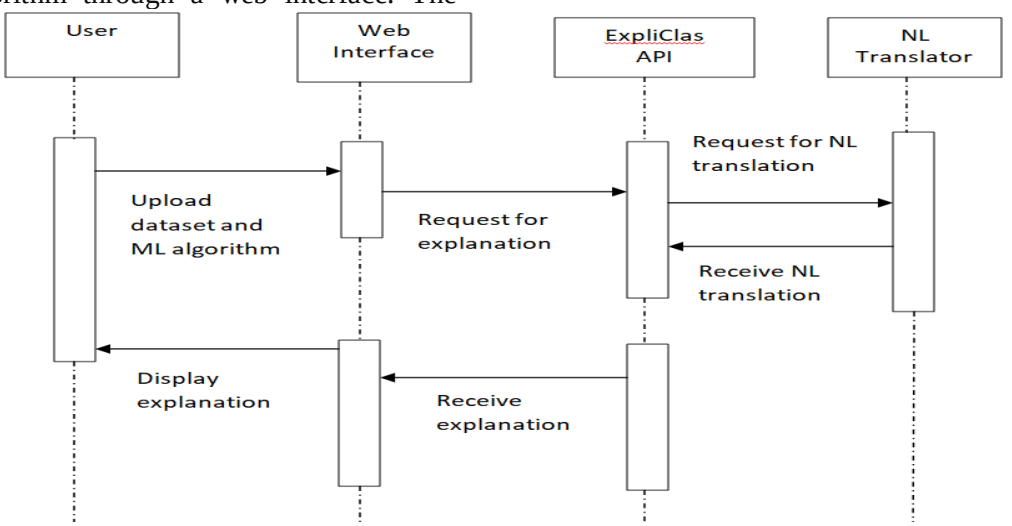

FIGURE 3. A SEQUENCE DIAGRAM FOR EXPLANTION PROCESS

2.4. *Data Flow Process in ExplainEx*

The flow of control for obtaining explanation is illustrated in figure 4. A user is first authenticated and the user

information preserved in memory. The user then starts a session generating a session id and time stamp. The user then selects a dataset and the classification algorithm. The dataset/classifier pair is then forwarded to the explanation engine. This process finally yields the required explanation which in turn is stored in the user profile.

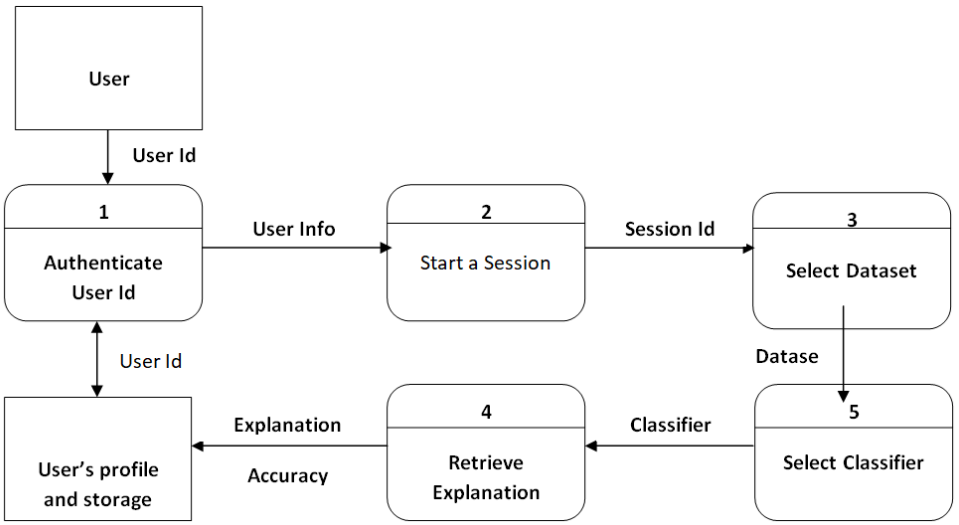

FIGURE 4. A DATA FLOW DIAGRAM FOR EXPLANATION PROCESS

3. THE EXPLAINEX SOFTWARE ARCHITECTURE

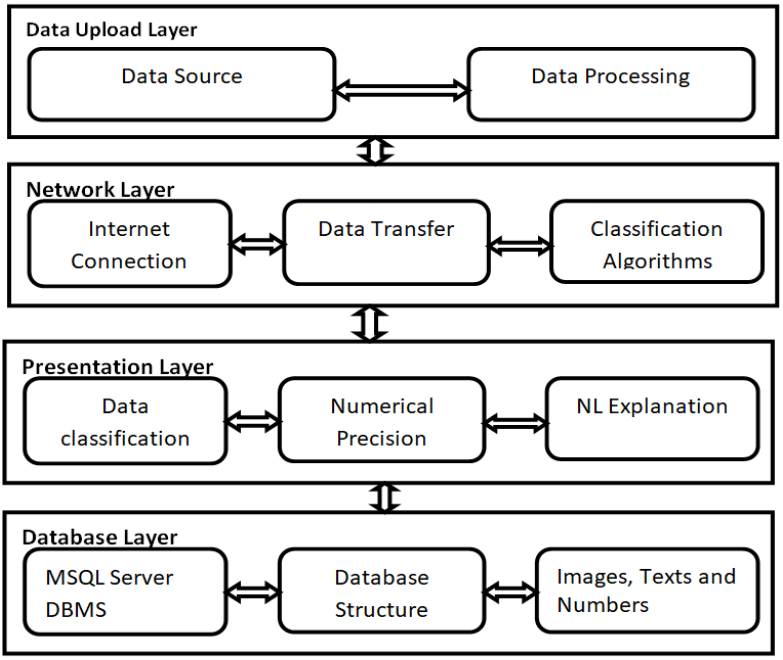

FIGURE 5. EXPLAINEX SOFTWARE ARCHITECTURE

3.1. The Main Components of ExplainEx Architecture

The software architecture consists of five main layers namely, data upload, network, presentation and database layers as illustrated in figure 5.

- **Data upload layer**

  The data upload layer consists of the data source and data processing modules. The data source module ensures that a dataset is selected and transported into the system. The displayed data is then analyzed to ensure compliance with formatting standards in order to prevent "Invalid data format" exception being thrown at other stages.

- **Network layer**

  The network layer ensures that a consistent connection is established between the host computer and the cloud server which hosts the API for explanation generation. The internet connection module establishes the required internet connection and monitors to ensure that this connection remains through the period of transaction. The named dataset is transferred to the cloud server through this module. This module also ensures that no duplicate data is uploaded in order to avoid "Duplicate data" exception being thrown by the application. This module ensures that the selected algorithm is transmitted to the cloud server to be processed in conjunction with the dataset.

- **Presentation layer**

This module displays the Data classification, Numerical precision and NL explanation for user viewing.

- **Database layer**

This module consists of the DBMS, Database structure, Images, texts, numbers configure in MS SQL server.

3.2. Algorithm Development

The ExplainEx software was designed from a systematically structured algorithm comprising of four sub functions in addition to the main function.

```
Begin
        Sid := StartSession

        If  Sid is true then
                Input Select dataset DN
                US  :=  UploadModel  (DN,
Sid)
        End if

        If  US is true then
                Input              Select
classification algorithm CN
                BS   :=   BuildClassifier
(DN, CN, Sid)
        End if

        If  BS is true then
                ES   :=   GetExplanation
(DN, CN, Sid)
        End if
End

Function StartSession
        If   there   exists   internet
connection then
                Sid := httpdownload(URL)
        End if
        Return Sid
End function
Function UploadModel (DN, Sid)
        //Confirm that DN is not empty
        Do While US is nothing or J is
less than 3
                US  :=  httpupload  (DN,
Sid)
                J++
        Enddo
        Return US
End function
Function BuildClassifier (DN, CN, Sid)
        //Confirm that CN is not empty
        BS := httpdownload (DN,CN,Sid)
        Return BS
End function

Function GetExplanation (DN, CN, Sid)
        ES  :=  httpdownload  (DN,  CN,
Sid)
        Return ES
End Function
```

Description of Variables Used in the Algorithm

Sid: Session id
DN: Name of dataset
US: Status of the upload
CN: Name of classification algorithm
BS: Status of build action
ES: Status of explanation action

4. GENERATING EXPLANATION

ExplainEx generates actual natural language explanations. It is made of two stages. The first stage provides the prediction accuracy of the algorithm used, which is presented in figures in final format. This gives an idea as to the level of precision at which the classification operates and serves to boost in trust a user has on the entire system. The second stage is the provision of natural language format of the confusion matrix and internal classes and their interpretation. In its general form the natural language form of the explanation will look similar across all the four algorithms. This is because the internal components of the model (confusion matrix and classes) remain the same. However, the precision figures are expected to be different because they are dependent primarily on the selected algorithm. Each algorithm has its own strength and weaknesses depending on the type of data and problem to be solve.

4.1. Configuration of FURIA Algorithm

The configuration of FURIA algorithm consists of ten parameters configured to produce maximum impact on the outcome of the explanation.

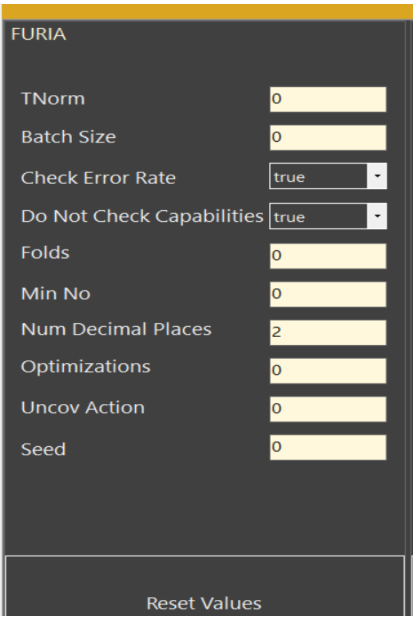

FIGURE 6. FURIA CONFIGURATION SCREEN

As illustrated in figure 4.6 the FURIA algorithm is configure based on the parameters TNorm, Batch size, Check error rate, Do Not Check Capabilities, Folds, Minimum number, Number of decimal places, Optimization, Uncover action and seed. These parameters are set within a certain numerical range apart from the "check error rate" and "Do not check capabilities" which are binary values denoted as yes/no.

4.2. Configuration of J48 Algorithm

The configuration of J48 algorithm is based on twelve parameters.

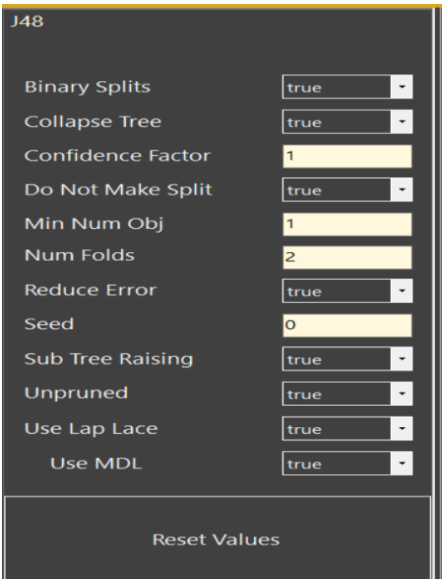

FIGURE 7. J48 CONFIGURATION SCREEN

As illustrated in figure 4.7, the twelve parameters that determine the output result from J48 algorithms are Binary splits, Collapse tree, Confidence factor, Do not make split, Minimum number of objects, Number of folds, Reduce error, Seed, Sub tree raising, Unpruned, Use Lap Lace and Use MDL. Most of these parameters are based on binary values yes/no and the rest are numerical values.

4.3. Configuration of RANDOM TREE Algorithm

The configuration status of Random Tree algorithm is determined by 10 parameters which are illustrated in figure 4.8.

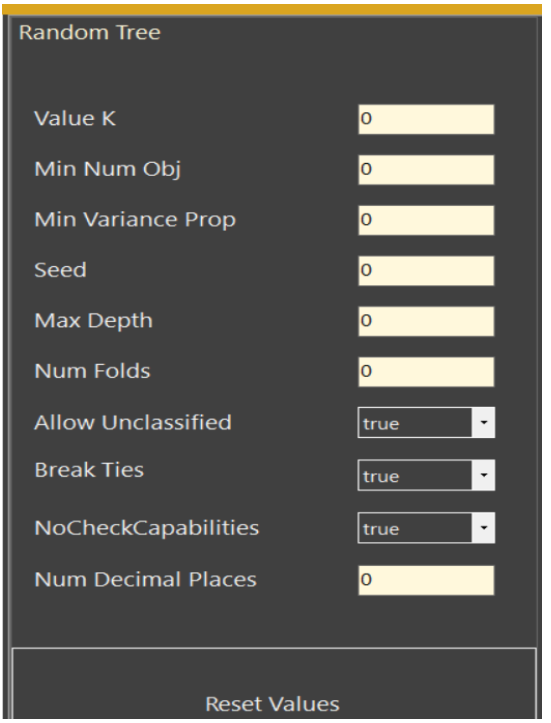

FIGURE 8. RANDOM TREE CONFIGURATION SCREEN

As clearly shown in figure 4.8, the ten parameters that control the behavior of the Random Tree algorithm are Value K, Minimum number of objects, Minimum variance property, Seed, Maximum depths, Number of folds, Allow unclassified, Break ties, No check capabilities, Number of decimal places. Like in the case of other algorithms, the parameters are grouped into binary values and numerical values.

4.4. Configuration of REP TREE Algorithm

The configuration of Rep Tree algorithm is dependent on ten parameters as specified in figure 4.9.

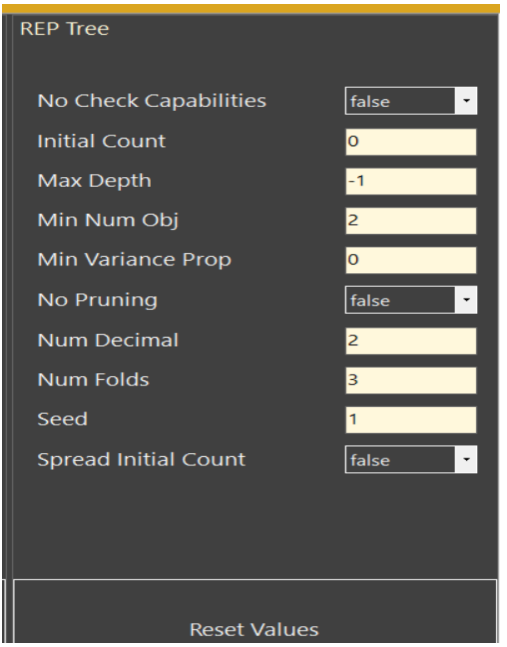

FIGURE 9. REP TREE CONFIGURATION SCREEN

As illustrated in figure 4.9 these parameters include No check capabilities, Initial Count, Maximum depth, Minimum number of objects, Minimum variance , Number of pruning, Number of decimal places, Number of folds, Seed and Spread initial count

5. SYTEM EVALUATION

There have not been total consensus in the standards for measuring explainability in AI systems (Eberle & Bundy, 2019). However, there are considerable levels of acceptability for some standards. Most researchers agree that XAI system can be said to have met its goal when the explanation is understandable, clear, efficient and interpretable (Hoffman, et. al., 2018; Amodei, et. al., 2016). To evaluate the ExplainEx system, two methods were adopted - use of existing state-of-the-art datasets to test the system, and use a trust Scale to measure user experience and satisfaction.

5.1. The Machine Learning Algorithms

To carry out the evaluation, four different machine learning algorithms (FURIA, J48, Random Forest and REPTree) were applied to seven unique datasets and their classification accuracy and global explanation observed in ExplainEx system. The datasets were adopted from the state-of-the-art WEKA open source data mining application which ships along with the distributable file (C:\Program Files\Weka-3-8\data\). The classification accuracy was found to be similar to those produced from WEKA on same datasets. The explanations were also found to reflect the original description of the data as shipped in the data folder. These results are shown in tables 1. The Trust Scale for measuring Explainable AI was adapted from (Jian,

1998). This scale seeks to find out directly from users whether they are confident in the XAI system, by measuring predictability, reliability, efficiency and believability. It contains eight questions, requiring the participants to choose any of the five options, strongly agree, I agree, I am neutral, I disagree, I strongly agree.

5.2. Evaluation Scale

The scale involved participants with extensive experience in the use of XAI systems. In this experiment, a total of twenty participants thoroughly used and evaluated the system. All the participants have either worked or currently working on AI and machine learning related projects. The items included in the scale are shown in table 2.The maximum point on the scale is five (5) and the lowest is one (1). Since there are eight (8) items on the scale, it goes to say that the highest score for a participant will be forty (40) and the lowest score (8). A majority of the items on the scale are adapted from Jian (1998), who also adapted the scale originally from Hoffman et. al.(2018) and Cahour (2010). The user survey result is presented in table 2.

TABLE 1. CLASSIFICATION AND EXPLANATION OF FDATASET

Dataset (IV): Glass				
SN	**Algorithm**	**Classification Accuracy**	**Original Data Description**	**Global explanation from ExplainEx**
1	FURIA	70.09%	Labeled values in attribute Type: build wind float, build wind non-float, vehic wind float, vehic wind non-float, containers, tableware, headlamps	There are 7 types of Glass: build wind float, build wind non-float, vehic wind float, vehic wind non-float, containers, tableware and headlamps. This classifier is quite confusing because correctly classified instances represent a 70.09%. There may be confusion related to most types of Glass.","Only in exceptional cases confusion involves vehic wind non-float.
2	J48	69.16%	Labeled values in attribute Type: float, non-float, wind float, wind non-float, containers, tableware, headlamps	There are 7 types of Glass: build wind float, build wind non-float, vehic wind float, vehic wind non-float, containers, tableware and headlamps. This classifier is quite confusing because correctly classified instances represent a 69,16%. There may be confusion related to most types of Glass. Only in exceptional cases

				confusion involves vehic wind non-float and tableware.
3	Random Tree	68.22%	Labeled values in attribute Type: float, non-float, wind float, wind non-float, containers, tableware, headlamps	There are 7 types of Glass: build wind float, build wind non-float, vehic wind float, vehic wind non-float, containers, tableware and headlamps. This classifier is quite confusing because correctly classified instances represent a 68.22%. There may be confusion related to some types of Glass. Specifically when we talk about types build wind float, vehic wind float, build wind non-float and headlamps.
4	REPTree	66.36%	Labeled values in attribute Type: float, non-float, wind float, wind non-float, containers, tableware, headlamps	There are 7 types of Glass: build wind float, build wind non-float, vehic wind float, vehic wind non-float, containers, tableware and headlamps.Th is classifier is quite confusing because correctly classified instances represent a 66.36%. There may be confusion related to most types of Glass. Only in exceptional cases confusion involves vehic wind non-float and containers.

TABLE 2. USER SERVEY RESULT

SN	ITEM	AVG	SD	VAR
1	I feel that the tool works and so I am confident using it	4.50	0.70	0.5
2	The tool produces predictable output	4.12	0.33	0.10
3	The tool produces reliable results	3.87	0.33	0.10
4	I feel safe using the results from the tool	4.37	0,48	0.23
5	The tool completes tasks quickly, it is efficient	4.37	0.48	0.23
6	The tool can do better than a novice human being	4.37	0.48	0.23
7	I will like to use the tool in making decisions	4.25	0.48	0.23
	Average User Satisfaction	**4.26**	**0.46**	**0.23**

5. CONCLUSION

In this paper the implementation of a prototype application for explaining predictive systems was presented, using four machine learning algorithms as use case. The application was tested using datasets from the distributable WEKA database. All the classifiers returned

precision level comparable to that of WEKA and the global explanation represented components from the original dataset. The application was also tested by eight participants currently working on machine learning projects and average user satisfaction rate of 4.26 was reported on a Hoffman Trust Scale of 5. The future research direction for this paper is twofold. First is to extend the explanation framework to LOCAL explanation. Secondly, to include other machines learning algorithms that are not decision tree based.

REFERENCES

Alexander, A., Jiang, A., Ferreira, C., & Zurkiya, D. (2019). An Intelligent Future for Medical Imaging : A Market Outlook on Arti fi cial Intelligence for Medical Imaging. *Journal of the American College of Radiology*, *17*(1), 165–170. https://doi.org/10.1016/j.jacr.2019.07.019

Amamou, S., & Cheniti-belcadhi, L. (2018). ScienceDirect ScienceDirect Systems Learning Tutoring In Tutoring In Project-Based Learning. *Procedia Computer Science*, *126*, 176–185. https://doi.org/10.1016/j.procs.2018.07.221

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). *Concrete Problems in AI Safety*. *277*(2003), 1–21. http://arxiv.org/abs/1606.06565

Calvaresi, D., & Främling, K. (2019). *Explainable Agents and Robots : Results from a Systematic Literature Review*. *Aamas*, 1078–1088.

Constantinescu, C., Stief, P., Dantan, J., Etienne, A., & Siadat, A. (2019). ScienceDirect ScienceDirect ScienceDirect Application potentials of artificial intelligence Application potentials artificial intelligence for the design of innovation processes for the design of innovation processes A new methodology to analyze the funct. *Procedia CIRP*, *84*, 810–813. https://doi.org/10.1016/j.procir.2019.04.230

Eoin, M., Mark, T., Kenny, E. M., & Keane, M. T. (2019). *Twin-Systems to Explain Artificial Neural Networks using Case-Based Reasoning : Comparative Tests of Feature-Weighting Methods in ANN-CBR Twins for XAI*.

Eberle, W., & Bundy, S. (2019). *Infusing domain knowledge in AI-based " black box " models for better explainability with application in bankruptcy prediction*.

Falade, A., Azeta, A., Oni, A., Odun-ayo, I. (2019). Systematic literature review of crime prediction and data mining. *Review of Computer Engineering Studies*, Vol. 6, No. 3, pp. 56-63. Published November 2019. https://doi.org/10.18280/rces.060302.

Hoffman, R. R., Mueller, S. T., Klein, G., & Litman, J. (2018). *Metrics for Explainable AI: Challenges and Prospects*. 1–50. http://arxiv.org/abs/1812.04608

Ibrahim, A., Gamble, P., Jaroensri, R., Abdelsamea, M. M., Mermel, C. H., Chen, P. C., & Rakha, E. A. (2020). Arti fi cial intelligence in digital breast pathology : Techniques and applications. *The Breast*, *49*, 267–273. https://doi.org/10.1016/j.breast.2019.12.007

Jian, J.-Y. (1998). *Foundations for Empirically Determined Scale of Trust in Automated Systems*.

Jiao, P., & Alavi, A. H. (2019). Geoscience Frontiers Arti fi cial intelligence in seismology : Advent , performance and future trends. *Geoscience Frontiers*, *September*. https://doi.org/10.1016/j.gsf.2019.10.004

Lamy, J., Sekar, B., Guezennec, G., Bouaud, J., & Séroussi, B. (2019). Arti fi cial Intelligence In Medicine Explainable arti fi cial intelligence for breast cancer : A visual case-based reasoning approach.

Artificial Intelligence In Medicine, *94*(January), 42–53. https://doi.org/10.1016/j.artmed.2019.01.001

Malgieri, G. (2019). *Automated decision-making in the EU Member States : The right to explanation and other " suitable safeguards " in the national legislations*. *35*. https://doi.org/10.1016/j.clsr.2019.05.002

Sharma, G. D., Yadav, A., & Chopra, R. (2020). *Artificial intelligence and effective governance : A review , critique and research agenda*. *2*(November 2019), 0–5. https://doi.org/10.1016/j.sftr.2019.100004

Siems-anderson, A. R., Walker, C. L., Wiener, G., Iii, W. P. M., & Haupt, S. E. (2019). Transportation Research Interdisciplinary Perspectives An adaptive big data weather system for surface transportation ☆. *Transportation Research Interdisciplinary Perspectives*, *3*, 100071. https://doi.org/10.1016/j.trip.2019.100071

Stoel, B. C. (2019). *Artificial intelligence in detecting early RA*. *49*, 25–28. https://doi.org/10.1016/j.semarthrit.2019.09.020