https://doi.org/10.2352/EI.2023.35.1.VDA-404

BioChipVis:An Information Visualisation Interface for Explainable Biochip Data Classification

Paul Craig, Department of Computing, Xi'an Jiaotong-Liverpool University, Suzhou, China
Ruben Ng, Department of Computing, Xi'an Jiaotong-Liverpool University, Suzhou, China
Yu Liu, Department of Computing, Xi'an Jiaotong-Liverpool University, Suzhou, China
Boris Tefsen, Molecular Microbiology, Utrecht University Padualaan 8, 3584 CH Utrecht, Netherlands
Sam Linsen, SquaredAnt Network Technology Co., Ltd., Hangzhou 311199, China

Abstract

This paper proposes a novel information visualisation interface to help with the reading and improvement of biochips. The interface serves two main groups of end users. These are bio-chip model users and bio-chip model developers. Bio-chip model users are biologists who use the software to read chips and detect biochemical substances. Bio-chip model developers use the software to design and train classification models by seeing how well the different biosensors work and how well the data fits their model. The interface proposed uses a Random Forest classifier and visualises the classification to provide a better understanding of how the data is classified by showing how it fits different classifications and how changes in attribute values can affect the classification. The interface also allows model-developers to interact to see how their model works for different attribute values, and shows them how new data (sent by model-users) fits into their classification model. This allow the biochip designers to detect how their model may be limited so they can retrain the model accordingly. The particular challenge with this project is how we manage and visualise uncertainty related to bio-sensor readings (that can be resultant from the manufacturing process and environmental factors) and the machine learning models, so that biologists can account for this when designing or using chips. Overall, our interface demonstrates the potential of information visualisation to be used to allow developers and model-users to better understand the effectiveness of classification models for their data, as well as the potential of collaborative interfaces to help them work together to build more effective supervised classification models.

Introduction

Machine learning algorithms for supervised classification, which learn from labelled data to classify new data, have proven to be effective in a number of important application areas such as detecting credit card fraud [1, 2], diagnosing medical conditions [3, 4], and image recognition [5, 6]. These classifiers are often cited as being effective as human-beings for classification task and have been shown to perform as well as humans, or even outperform humans, in a number of benchmark tests [7, 8]. Machine learning algorithms, and computers in general, are very effective when it comes to *remembering* and rapidly processing vast amounts of data. These are important abilities for classification and decision-making tasks in general, but they are by no means the whole story. Indeed, computers are somewhat less effective at other aspects of decision-making such as drawing on diverse sources of data, adapting to new patterns, reasoning with data, or acting in an appropriate informed manner once decisions are made [8, 9]. For these reasons much of the attention in the area of machine-learning is moving away from improving algorithm performance into the area of *explainable AI* [10] which looks at how users can become a more informed part of the machine learning and associated decision making process.

An issue with explainable AI is that the most effective models tend to act as black-boxes including large numbers of calculations. This makes it difficult to explain their operation effectively to end-users [11]. Machine learning models also tend to depend heavily on the quality and completeness of the training data. Specifically, a supervised classification model cannot be relied on to effectively classify new data correctly if it has not already seen that particular type of data labelled correctly in the training dataset. This can be a problem when a machine learning model is used *in the wild* in real word situations to classify data that is not similar to the data it was trained with [12]. Such a model will still give *a* classification for this data, but the classification will not be reliable and (perhaps more significantly) there will not be no indication than classification is not reliable. This can be a serious problem if decisions (such as the decision of a bank to give someone a loan, or the decision of a doctor to treat a patient) are to be based on such unreliable classifications. If a model is prone to give these sorts of unreliable classifications then the problem is more serious, so it is well worth considering how to avoid this issue.

The work presented in this paper investigates how we can improve the process of supervised classification by using an interactive information visualisation technique that gives users an overview of the model output showing how changes in attribute values change the classification result. The technique is also designed to be collaborative, which allows model-developers and model-users to detect and resolve issues related to a limited training data-set. Our visualisation gives model-users an indication of how reliable the classification model is for their data and reports any new data being classified to the model-developers so they can have an overview of how complete their training data-set is and how effective their model is. This would help the model-developers to tell if their model needs to be re-trained so that an improved model can be re-distributed to model-users.

Requirements Analysis

Requirements analysis for our application was performed with a team of six biologists involved in the early stages of developing a supervised learning model for the automatic classification of bio-chips designed to detect different types of antibiotic pollution [13, 14, 15]. This allowed us to identify two distinct user groups for collaborative supervised learning. These two groups are described as follows.

1. *Model-users*. These would be end users of the bio-chips who need to use the supervised-classification model to classify new data. This group has the following requirements.
   (a) See which class their data is most likely to belong to.
   (b) Know how reliable the classification is.
   (c) Know what other potential classifications are more or less probable.
   (d) Know how changes to their data would result in different outcomes.
2. *Model-developers*. These would be the biologists responsible for developing the bio-chips and bioinformaticians responsible for training the model to classify new samples. This group has the following requirements listed below.
   (a) Evaluate the overall effectiveness of the model.
   (b) Evaluate the effectiveness of the model for different classes.
   (c) Evaluate the effectiveness of the model for different patterns in the data.

Subsequent meetings with the biologists allowed us to characterise the data generated by the bio-chips as multidimensional data with numeric attributes. Each bio-chip would be covered in an array of spots that change colour according to different bio-chemical reactions (so that patterns of colour could be used to identify different types of antibiotic pollution after the chip is exposed to a sample). After the chips are scanned the colours can be converted to numeric values so that each chip has an array of values that can be used to determine the type of antibiotic pollution it has been exposed to (see Figure 1). Some of the spots on the chip will be biological replicates that can have their values combined, so the actual number of values generated from each chip will be smaller than the number of spots (around five or six). The chips can also be classified as belonging to either one of five different classes. These classes relate to the absence of antibiotics or either of the four common types of antibiotic. So, our data could be characterised as multidimensional data with a relatively small number of five or six numeric attributes with each instance belonging to one of five different classes.

This general data type is relatively common and is one of the seven data types identified in Shneiderman's task by data type taxonomy for information visualisation [16]. We can also characterise our data as having a relatively small number of classes and a small number of attribute values which are all numeric.

Related work

If we look at the requirements of our biologists we can see that some of these can already be met effectively by existing techniques, and some can not. The first model-user requirement is to see which class their data is likely to belong to (section , requirement 1a). All supervised learning algorithms for classification will at least assign data to at least one class to satisfy this requirement.

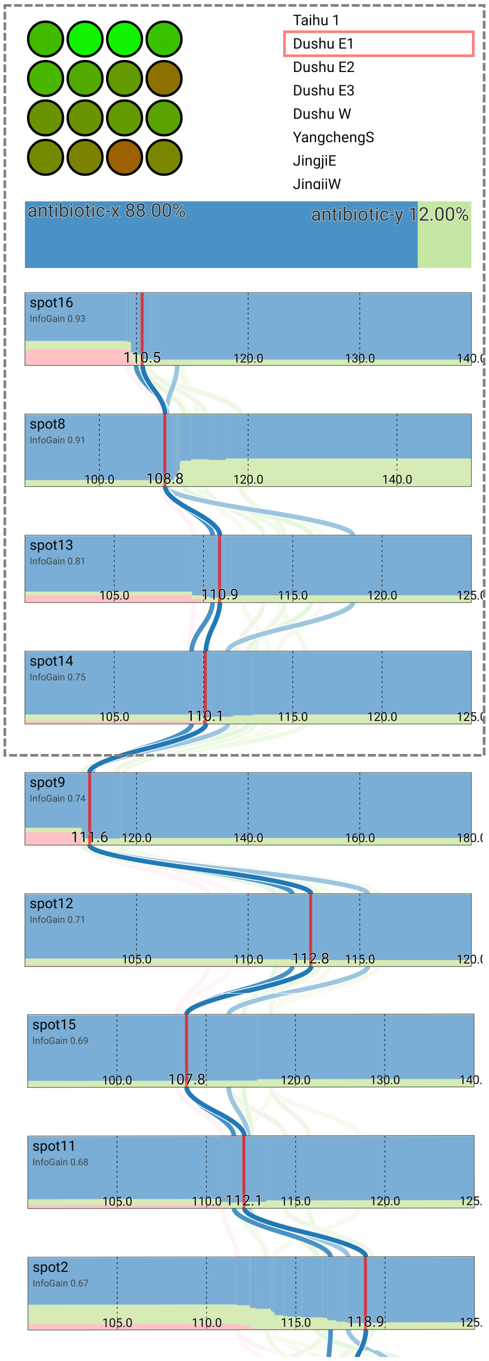

Figure 1. *The smartphone interface showing the results of scanning a biochip with the classification and contribution of different spot values. The four spot values that have the greatest information gain to affect the classification are on the screen initially with the user being able to scroll down to see more.*

Our second model-user requirement (requirement 1b), to know how reliable the classification is with relation to other potential classifications, can be satisfied by algorithms (such as Neural Networks [17] and Random Forest classifiers [18]) that assign a value (or score) to each class for the given data. These values can be used as an indication of uncertainty in the classification [19] and an indication of the probability that the data belongs to another class. For example, if a Random Forest classifier outputs a score of 15 for Class A and a score of 14 for Class B, the predicted class will be A but we will also have an idea that there is some uncertainty in the prediction and a likelihood that Class B could also be a possible candidate for the data provided.

Our third model-user requirement (requirement 1c) is to know how changes to their data would result in different outcomes. This is important for our own case study of antibiotic pollution detection, as the values generated by scanning the colour of individual spots on our bio-chips may not be accurate due to external factors impeding one of the biochemical reactions. In this type of situation a small change in the spot value could cause the data to be classified in a different way and the biologist may want to use a new chip or some other method to confirm the classification result. For other types of classification, changing input values could be helpful to help the model-user understand how the data affects the result of the model classifications. For example, if a model is used to predict a patient's likelihood of being diagnosed with a disease, changing input values such as the patient's weight or minutes of daily exercise could help them understand how life-style choices affect their health.

One way to let model-users know how changes to their data result in different outcomes is to allow them to change values in their data and reclassify the data to view the result [20, 21]. Interactive visualisations that allow us to rerun classifiers with different values and show the results, satisfy this requirement to a certain extent. For example, the RuleMatrix visualisation allows users to interact to filter a set of data that is classified by a rule-based model bases on a neural network [21]. These techniques do not however explicitly show the result of changes until the user interacts with the visualisation to see the result. This means that important information such as the sensitivity of attributes to value changes may not be apparent.

Other techniques, allow the users to see how the data is classified by showing the decisions used in the classification process [21, 22, 23]. These types of technique tend to use rule based classifiers such as decision trees which are considered to be more *explainable* and *transparent* as the process applied by the model to classify the data are relatively easy for human beings to comprehend [23]. The drawback of these types of technique is that they don't give any indication of the uncertainty in the classification and can perform badly with data where patterns are more complex. This contrasts with *black-box* classifiers such as Neural Networks and Random Forest classifiers which tend to perform well will all types of data, but are often referred to as being *opaque* and having poor explainability due to the large number of calculations they use to arrive at a result [17, 18] .

The requirements of our model-developers are also satisfied to varying degrees using existing techniques. The first model-developer requirement (requirement 2a) is to evaluate the overall effectiveness of the model. In order to evaluate the overall effectiveness of a supervised learning model, we can measure its accuracy with a test data-set that is distinct from the training data. Accuracy is the number of elements that are correctly classified divided by the total number of elements and is represented as a number in the range from one (all elements correctly classified) to zero (no elements correctly classified). Usually a data-set is trained using a piece of a labelled data set that has been randomly chosen, and then assessed (for accuracy, etc.) using the remaining portion of the same labelled data set. (see Figure 2).

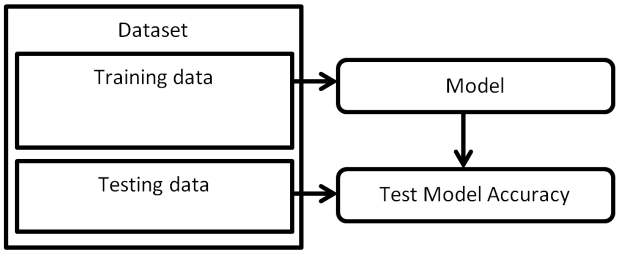

Figure 2. *Supervised learning models for classification are trained using a randomly selected portion of a labelled data-set and evaluated with the remainder of the same labelled data-set.*

Accuracy can also be calculated for individual classes together with precision and recall. Recall is the number of elements correctly classified as belonging to a particular class divided the total number of elements actually belonging to that class. Precision is the number of elements correctly classified as belonging to a particular class divided the total number of elements of any class classified rightly or wrongly as belonging to that class. Other metrics, such as the f-score, are a balance of these two metrics [24].

Uncertainty of data may appear because of the production environment and environmental factors. For example a doctor may diagnose the patient by analysing the medical data, such as blood pressure, blood test, and oxygen saturation. But by just seeing these data, it will cause uncertainty. Doctor may need to know patient's medical history, in order to diagnose correctly. In our case study, we consider that bio-chip also may result uncertainty that might caused by production, environmental factor (e.g. temperature, pH level) and the machine learning classifier. Therefore, we studied on how other researchers have done to display the uncertainty result.

There are different way to represent the uncertainty. The first two type categorised as static representation[25]; first, it can be differentiate with individual representations presented for an attribute and its linked uncertainty. Second, representations can be mixed, where a single visualization shows both an attribute and its uncertainty - an attribute and its uncertainty are displayed by overlaying one on top of the other using appropriate visual variables. The third method is visualising in an interactive computer environment[25], allow users to control the display of both the data and the uncertainty of the data.

The two general categories for techniques to represent uncertainty proposed by are intrinsic and extrinsic [26]. Intrinsic representation techniques display the uncertainty by differing an object's appearance, which include varying visual variables such as texture, brightness, hue, size, orientation, position, or shape. For example, finer texture or darker value could represent greater reliability and coarser texture and lighter values could represent uncertainty. To represent uncertainty on extrinsic techniques, it applied geometric objects such as arrows, bars, and complex structures (such as pie charts).

There are three types of uncertainty according to [27, 28].The first is statistical, which refers to the data's distribution, or the estimated mean and standard deviation (confidence interval). The second is error, which can be either estimates or disparities between a known correct datum and an estimate with an absolute value. The final term is range, which refers to the time span over which the data is available (and cannot be quantified into either the statistical or error definitions). In our case study, we proposed the prototype to represent the uncertainty interactively and add animation [29], which is a part of Human Computer Interaction (HCI). As our data is multidimensional data, the uncertainty can be multiplied and hard to quantify. We will make our value viewable, and able animate the classification result in the prototype.

In order to satisfy our model-developers second requirement to evaluate the effectiveness of the model for different classes (requirement 2b) our biologists could use another statistical tool known as a confusion matrix [30]. This provides a more detailed insight into a model's performance by showing how the actual class of elements map to the predicted class for every pair of classes. This is generally represented as a grid (with every class being represented as a row and column), and can be colour coded to highlight classes that are *confused* by the model.

Our biologists' third requirement, to evaluate the effectiveness of the model for different patterns in the data (requirement 2c), is less well served by existing techniques. While we can indicate how well the model works for patterns already present in the test data-set using the methods described above, these methods and metrics will not work for patterns that are not present in the test-data set and the model is indeed unlikely to work well to classify these patterns either. These *unknown* patterns can be thought of as gaps in the training data.

A simplified example of a problematic *gap* in the training data would be a model used to identify leafs according to their shape being trained and tested using wide-long leaves, narrow-long leaves and narrow-short leaves, but without wide-short leaves. If the model is then used to identify a wide-short leaf it is likely to erroneously classify it along with wide-long or narrow short leaves. If this problem leads to a leaf being miss classified, it's not likely to be too much of an issue. But in another domain this type of false-positive could lead to someone being refused a bank-loan or being prescribed unnecessary medication.

It is tempting to think that the problem of gaps in our training data could be avoided by simply using a larger training set that contains a wider range of samples. Indeed, if the training data-set was truly representative of the sample universe, then there would be no gaps in the training data and no problem. There are, however, a number of reasons why a truly representative sample may not be feasible. Firstly, the cost of building an truly exhaustive training set may be prohibitive. It may also be the case that the sample universe is growing and new patterns are emerging (e.g. in the case of using bio-chips to detect viruses when a new strain evolves). Therefore, it is necessary to consider how to manage the problem of potential gaps in the training data, rather than to think we can try and avoid the problem altogether.

The problem for model-developers trying to manage gaps in the training data is that its difficult to tell if gaps in the training data are just gaps in the training data, or if they reflect gaps in the sample universe. For example, if our training-data lacks wide-short leaves, it could be because our training data is not enough, or that their are actually no wide-short leaves in the world. Indeed, rather than rely on model-developers to identify gaps in the training data, it would be a lot easier to identify the *new* patterns when a model-user attempts to classify data that has that pattern. In order to preempt this problem, the application proposed in this paper allows users to identify that they have detected novel patterns in their data (i.e. patterns that cannot be classified effectively using the model) for this data to be sent back to the model-developers who can update the model accordingly.

Most of our complete set of user requirements can be satisfied using existing techniques. Machine learning models can be used to tell the model-users what class their data belongs to (requirement 1a) and techniques that score different classes for new data (such as Neural Networks and Random Forest classifiers) can give the user and indication of how reliable the classification is (requirement 1b) and what other classifications are more or less probable (requirement 1c). Metrics generated using a test data-set can show model-developers how effective their model is overall (requirement 2a) and confusion matrices can tell the model-developer how effective their model is at differentiating between different classes (requirement 2b). Existing techniques are somewhat less effective at showing model-users how changes in attribute values affect the classification result (requirement 1d) or showing model-developers how effective their model is for different patterns in the data (requirement 2c), particularity if these patterns are not present in the data the model is trained with. The technique developed for this paper is designed to satisfy these two user requirements for data with smaller numbers of numeric attributes and smaller numbers of classes as described in section .

BiochipVis

The application introduced in this paper is an interactive mobile interface that allows users to classify biochips readings to show the classification changes for adjustment in each value and allows input values to be adjusted with the representation recalculated quickly. Therefore, users can see the changes on the data classification and display how the classification is to changes in individual bio-sensor values. The interface uses a Random Forest classifier and the machine learning model is able to predict imprecise measurements where a small difference in any input value can potentially change the output of the model.

The *results-panel* shows the classification result for the current attribute values by showing the percentage of the total classification score given to each separate class. This uses a wide rectangle divided into colour coded segments with their width proportional to percentage for each class. The segments are labelled and colour codded according to a colour-scheme adapted from the Brewer colour schemes for accessibility [31].

The central *attributes-panel* has wide rectangles *Attribute boxes* that show the users which attribute values are selected and how changes in value change the classification result. These are ordered according to the Information Gain for training-data attribute values so that the attributes most capable of separating the training data into different classes are closest to the top. For each attribute box, a red line is used to show the currently selected value. The boxes can also act like stacked bar charts [32] showing the relative score for each class for each attribute value assuming that the values for all other attributes remain constant. The user can also click and drag on the red line so see how the classification changes with different values. When this happens the red line moves gradually to follow the mouse cursor so as the tran-

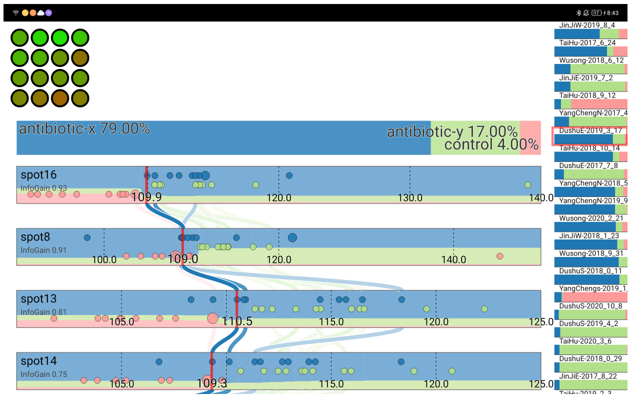

Figure 3. *Classification of sample DushuE 2019-31-7*

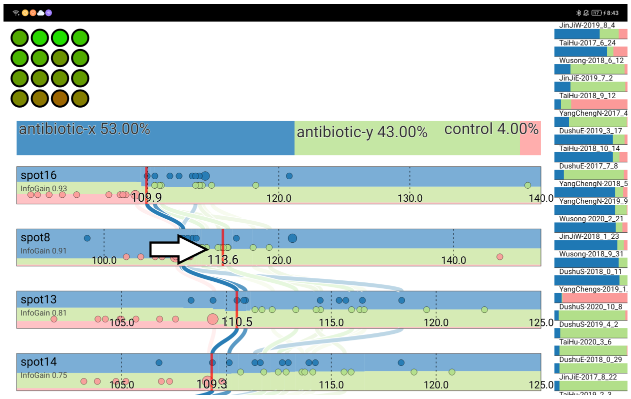

Figure 4. *Classification result changes after adjusting the value for spot 8.*

sition between views is smooth [33, 34]. This allows the user to see how sensitive the classification is to changes in the values for individual attributes.

The curved lines in the central attribute panel act like a parallel coordinates plot [35, 36] to show the values for instances in the training data-set closest to the currently selected values. The weight of the colour of each line is proportional to the euclidean distance of the training data instance to the selected values in normalised attribute space so that more similar training instances appear darker and less similar instances fade into the white background. This shows the user what training-data is similar to their own data and gives them an indication of how well the model is suited to classifying their data.

Users can manipulate to explore the sensitivity of their classification to changing different spot values. This can help user to understand the classification and better comprehend uncertainty in their data. For example, by selecting the sample DushuE 2019-31-7 (Figure 3), we can see the classification antibiotic x may be sensitive to changes in the value for spot 8. By clicking and dragging to adjust this spot value (with the red line and classification animating gradually to a new hypothetical value) we can see the potential change in the confidence in the classification toward antibiotic y (Figure 4).

Evaluation

We performed an initial evaluation using the cognitive walkthrough method [37, 38], which is a usability evaluation process that focuses on a user's cognitive activities. The evaluation was done with a small number of model-users where each participant was asked to explore the data and describe patterns they were able to find. During the evaluation the users were able to identify a number of significant patterns related to imprecise readings that they would not be able to identify using other techniques. Examples of this were the levels of uncertainty around readings of 115 on spot 8 and 110 on spot 16. Small changes around these values would often change the confidence in a classification to a considerable degree. The biologist let us know that this information was useful and might prompt them to examine the physical chip, use another chip to confirm the reading, or consider modifying the design of the chip to be used.

Conclusion

We have developed an interactive information visualisation application to support with the reading and improvement of biochips classification for model-developers and model-users. The application is designed to work on smartphones for model-users and a tablet connected to a large display for model-developers. The visualisation shows the output of the model and how the model changes for different attribute values rather than showing the workings of the model. This allows model-users to gauge uncertainty for their classification and see how their classification might be sensitive to changes in attribute values. Model-developers can explore to have an overview of how their model performs and see how new data fits their model. This should allow model-developers to find the limitations in their model and retrain it accordingly. A small-scale user test was performed in order to test the application interfaces which both scored well for different usability metrics.

The interface currently works with a random-forest classifier on up to sixteen numeric attributes (with four main attributes onscreen and others access-able through scrolling) and three or four classifications with around fifty or a hundred data-instances in the training data-set. The model-developer interface is designed to be used on more powerful devices so it could conceivably handle larger scale-data with more instances. As the visualisation displays the classifier output rather than the workings of the classifier itself, it should be able to display any sort of classifier for this sort of data. However, performance and interface response time are likely to be a problem on smartphone devices for more demanding classifiers such as neural networks. Especially as the interface needs to recalculate numerous output values while attribute values are adjusted by the user. Larger data sets would also be an issue for the interface as the number of classes is limited by the number of colours the user differentiate and the number of attributes is limited by the vertical space on the devices screen. A design for larger data with more attributes or classes could possibly work by folding classes together or automatically reordering attributes according to their relevance.

According to our user evaluation, results for the data-sets used demonstrate great potential for this sort of technique for collaborative information visualisation driven explainable machine-learning. This type of visualisation is likely to have more potential applications as the use of machine learning becomes more prevalent and mobile device capabilities improve [39, 40] with developers also coming round to see the advantage of making machine learning more explainable and accessible [9, 10]. For future work we plan to develop data collection functionality into the model-users' smartphone application and test the two types of applications working together to scan and classify bio-chips for antibiotic pollution monitoring.

Acknowledgment

This project is supported by Key Program Special Fund XJTLU Project KSF-E-10.

References

[1] Y. Kultur and M. U. Caglayan, “Hybrid approaches for detecting credit card fraud,” *Expert Systems*, vol. 34, no. 2, pp. 1–13, 2017.

[2] N. K. Mendhe, M. N. Thakare, and G. D. Korde, “Supervised machine (svm) learning for credit card fraud detection,” *International Journal of Engineering Trends & Technology*, vol. 8, no. 3, pp. 137–139, 2014.

[3] J. A. Ozolek, A. B. Tosun, W. Wang, C. Chen, S. Kolouri, S. Basu, H. Huang, and G. K. Rohde, “Accurate diagnosis of thyroid follicular lesions from nuclear morphology using supervised learning,” *Medical Image Analysis*, vol. 18, no. 5, pp. 772–780, 2014.

[4] A. Masood, A. Al-Jumaily, and K. Anam, “Self-supervised learning model for skin cancer diagnosis,” in *International IEEE/EMBS Conference on Neural Engineering*, 2015.

[5] R. Gross, L. Sweeney, F. D. L. Torre, and S. Baker, “Semi-supervised learning of multi-factor models for face de-identification,” in *IEEE Conference on Computer Vision & Pattern Recognition*, 2008.

[6] S. N. Srihari, “Semi-supervised learning for handwriting recognition,” in *International Conference on Document Analysis & Recognition*, 2009.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] E. Brynjolfsson and T. Mitchell, “What can machine learning do? workforce implications,” *Science*, vol. 358, no. 6370, pp. 1530–1534, 2017.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, “" why should i trust you?" explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[10] D. Gunning, “Explainable artificial intelligence (xai),” *Defense Advanced Research Projects Agency (DARPA), nd Web*, vol. 2, 2017.

[11] F.-Y. Tzeng and K.-L. Ma, *Opening the black box-data driven visualization of neural networks*. IEEE, 2005.

[12] K. Allix, T. F. Bissyandé, Q. Jérome, J. Klein, Y. Le Traon *et al.*, “Empirical assessment of machine learning-based malware detectors for android,” *Empirical Software Engineering*, vol. 21, no. 1, pp. 183–211, 2016.

[13] L. Gang, Y. Zhiyong, T. Xiuyi, and C. Junfeng, “Advances of detection technologyng for antibiotics from livestock and poultry breeding wastewater,” *Journal of Green Science and Technology*, no. 11, p. 54, 2011.

[14] D. I. Andersson, “Persistence of antibiotic resistant bacteria,” *Current opinion in microbiology*, vol. 6, no. 5, pp. 452–456, 2003.

[15] J. Björkman, D. Hughes, and D. I. Andersson, “Virulence of antibiotic-resistant salmonella typhimurium,” *Proceedings of the National Academy of Sciences*, vol. 95, no. 7, pp. 3949–3953, 1998.

[16] B. Shneiderman, “The eyes have it : A task by data type taxonomy for information visualization,” in *Proc IEEE Symposium on Visual Languages*, 1996.

[17] M. H. Hassoun *et al.*, *Fundamentals of artificial neural networks*. MIT press, 1995.

[18] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, “How many trees in a random forest?” in *International workshop on machine learning and data mining in pattern recognition*. Springer, 2012, pp. 154–168.

[19] G. Montavon, W. Samek, and K.-R. Müller, “Methods for interpreting and understanding deep neural networks,” *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[20] F. Grün, C. Rupprecht, N. Navab, and F. Tombari, “A taxonomy and library for visualizing learned features in convolutional neural networks,” *arXiv preprint arXiv:1606.07757*, 2016.

[21] Y. Ming, H. Qu, and E. Bertini, “Rulematrix: Visualizing and understanding classifiers with rules,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 342–352, 2018.

[22] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mane, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg, “Visualizing dataflow graphs of deep learning models in tensorflow,” *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 1–12, 2017.

[23] T. D. Nguyen, T. B. Ho, and H. Shimodaira, “A visualization tool for interactive learning of large decision trees,” in *Proceedings 12th IEEE Internationals Conference on Tools with Artificial Intelligence. ICTAI 2000*. IEEE, 2000, pp. 28–35.

[24] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 161–168.

[25] A. M. MacEachren, *How maps work: representation, visualization, and design*. Guilford Press, 2004.

[26] N. Gershon, “Visualization of an imperfect world,” *IEEE Computer Graphics and Applications*, vol. 18, no. 4, pp. 43–45, 1998.

[27] A. T. Pang, C. M. Wittenbrink, S. K. Lodha *et al.*, “Approaches to uncertainty visualization,” *The Visual Computer*, vol. 13, no. 8, pp. 370–390, 1997.

[28] M. Riveiro, “Evaluation of uncertainty visualization techniques for information fusion,” in *2007 10th International Conference on Information Fusion*. IEEE, 2007, pp. 1–8.

[29] C. Gonzalez, *Animation in user interface design for decision-making: a research framework and empirical analysis*. Texas Tech University, 1995.

[30] J. Parker, “Rank and response combination from confusion matrix data,” *Information fusion*, vol. 2, no. 2, pp. 113–120, 2001.

[31] M. Harrower and C. A. Brewer, “Colorbrewer. org: an online tool for selecting colour schemes for maps,” *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003.

[32] Q. Alsbury and D. Becerra, “Displaying stacked bar charts in a limited display area,” Aug. 7 2012, uS Patent 8,239,765.

[33] P. Craig, N. R. Seïler, and A. D. O. Cervantes, “Animated geo-temporal clusters for exploratory search in event data document collections,” in *2014 18th International Conference on Information Visualisation*. IEEE, 2014, pp. 157–163.

[34] P. Craig and N. Roa-Seïler, “A vertical timeline visualization for the exploratory analysis of dialogue data,” in *2012 16th International Conference on Information Visualisation*. IEEE, 2012, pp. 68–73.

[35] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu, “Scattering points in

parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1001–1008, 2009.

[36] M. Graham and J. Kennedy, "Using curves to enhance parallel coordinate visualisations," in *Proceedings on Seventh International Conference on Information Visualization, 2003. IV 2003.* IEEE, 2003, pp. 10–16.

[37] J. Rieman, M. Franzke, and D. Redmiles, "Usability evaluation with the cognitive walkthrough," in *Conference companion on Human factors in computing systems*, 1995, pp. 387–388.

[38] N. Ghasemifard, M. Shamsi, A. R. R. Kenari, and V. Ahmadi, "A new view at usability test methods of interfaces for human computer interaction," *Global Journal of Computer Science and Technology*, 2015.

[39] P. Craig, "Interactive animated mobile information visualisation," in *SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications*, 2015, pp. 1–6.

[40] P. Craig, X. Huang, H. Chen, X. Wang, and S. Zhang, "Pervasive information visualization: toward an information visualization design methodology for multi-device co-located synchronous collaboration," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 2015, pp. 2232–2239.