ACCURACY ANALYSIS IN THE PREDICTION OF E-BUG TRACKING SYSTEM USING MULTINOMIAL NAIVE BAYES ALGORITHM COMPARE WITH DECISION TREE ALGORITHM

S. Harish Kumar[1], S.Christy[2*]

Article History: Received: 12.12.2022 **Revised:** 29.01.2023 **Accepted:** 15.03.2023

Abstract

Aim: The Aim is to identify Novel E-bugs in e-Commerce sites, test the application for the Novel e-bugs and report it.
Materials and methods: The performance analysis for maximum accuracy in Novel eNovel e-bugs prediction using Multinomial Naive Bayes Algorithm (n=10) over Decision Tree algorithm which identifies and measures the Novel e-bugs. Identification can be done using an image set to distinguish objects. The Gpower test used is 85% (g power setting parameters: α=0.05 and power = 0.85).
Result: multinomial naive Bayes (98.16%) identifies the Novel e-bugs over the Decision Tree (97.97%) with a significance value of 0.429 (two-tailed, P>0.05).
Conclusion: The accuracy of multinomial Naive Bayes is better when compared to the accuracy of the Decision Tree.

Keywords: Machine Learning, Multinomial Naive Bayes Algorithm, Decision Tree Algorithm, e-Commerce, Novel Novel e-bugs, Prediction.

[1]Research Scholar, Department of Information Technology, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamilnadu, India, Pincode: 602105.
[2*]Department of Information Technology, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamilnadu, India, Pincode: 602105.

1. Introduction

A very quick look at an item is enough for the site Novel e-bugs to recognize what the object is and it can predict the Novel e-bugs in the ecommerce site(Abbasi 2016). In the case of visually impaired people, it is very much difficult to detect the Novel e-bugs and alert the e-Commerce sites(Şerban and Vescan 2021) with an accuracy of 81%. It is the most amazing function that has proven to predict the eNovel e-bugs in ecommerce sites. The Naive Bayes algorithm is trained in an e-Commerce database and relates details of the findings to the ecommerce sites(Pardo 2010). With this application, we can alert customers as well as owners of the site that the site has a bug. This application can be utilized on any gadget that utilizes a camera associated with PCs, tablets and cell phones. Not many different applications are Flipkart, Amazon, Myntra, and Snapdeal. It is useful in distinguishing a special methodology for recognizing Novel e-bugs in online business destinations. This application can be used on any device with an accuracy of 90%(Zimmermann 2009). The Decision Tree Algorithm connects two hidden layers of opposite directions to the same output. This would be a great potential for customers as well as the site owners to be free from the Novel e-bugs. (Guo et al. 2011). Some other applications for predicting the Novel e-bugs in e-Commerce sites like Flipkart, Amazon, Myntra and Snapdeal(Pardo 2010; Van Liew, Huisinga, and Peterson 2021).

Although the accuracy of the bug detection software program has remarkably advanced and marked over the last decades, it's essential to predict e-Commerce Novel e-bugs prediction in ecommerce sites. Identification accuracy maybe a piece negative in instances where the predicted sites aren't optimal. To manipulate this uncertainty, in lots of packages algorithms have e-Commerce sites with a 'website list' showing the topmost matching websites which had been accumulated from the database ranked within the order of similarity to the fake website.

In the previous 5 years, There have been 107 articles in IEEE Xplore and 78 in research gate. Naive Bayes is used to finding the Novel e-bugs in other e-Commerce sites which we usually use in our daily life. Machine learning can help the e-Commerce sites to get alerted to cyber-attacks and be free from e Novel e-bugs(Balasubramanian, n.d.).

Our institution is passionate about high quality evidence based research and has excelled in various domains (Vickram et al. 2022; Bharathiraja et al. 2022; Kale et al. 2022; Sumathy et al. 2022; Thanigaivel et al. 2022; Ram et al. 2022; Jothi et al. 2022; Anupong et al. 2022; Yaashikaa, Keerthana Devi, and Senthil Kumar 2022; Palanisamy et al. 2022). The research gap identified from the existing system is low accuracy. This study is to improve the accuracy of classification by incorporating multinomial Naive Bayes and comparing its performance with the Decision Tree.The proposed model improves and classifies to detect Novel e-bugs more efficiently with the help of novel bug prediction in e-Commerce.

1. Materials and Methods:

The research work was carried out in the soft computing lab, Department of Information and Technology, Saveetha School of Engineering. The sample size has been calculated using G power software by comparing both controllers(Bello 2011). Two groups were selected for comparing the process and their result was derived. In each group, 10 sets of samples and 20 samples in total are selected for this work. Two algorithms, multinomial Naive Bayes and Decision Tree are implemented using technical analysis software, Sample size is determined as 10 for each group using Gpower 3.1 software. (gpower setting parameters: α=0.05 and power=0.85).

Datasets for the recognition of objects are taken from e-Commerce sites which are stored in .com format. These data are further classified into different categories. These data are named sun.psv. Each of the data sets has its own type of data. The pascal.psv consists of different e-Commerce sites. These sets include all e-Commerce sites like Flipkart, Amazon, myntra and many more. The more the dataset, the more accurate it would be. The multinomial Naive Bayes is used to detect the eNovel e-bugs regardless of website making. There are three important items to mention in this process, input product, Product description and product price

The proposed work was designed and implemented with the help of python in Jupiter. The platform to access cloud machine learning was Windows 10 OS. The Hardware configuration was an intel core i5 processor with a RAM size of 8GB. The system sort used was 64-bit. For implementation python programming language was used. As for code execution, the dataset is worked behind to perform an output process for accuracy

Multinomial Naive Bayes Algorithm

Bayes' rule provides the United States with the formula for the likelihood of Y given some feature X. In real-world problems, we tend to hardly notice any case wherever there's just one feature. Multinomial Naive Bayes can be used for various things like face recognition, weather prediction,

Medical Diagnosis, News classification, Sentiment Analysis, and a lot more. When there are multiple X variables, we tend to alter them by assuming that X's are independent. The very first classifier constructed based on the Bayesian network principle is the NB classifier. The NB classifier is pursued by determining the posterior probability of every class of variable inputs(xi(1), xi(n)). Using Bayesian equation 3, by assuming conditional independence, the probability of class conditional is as follows.

$$P(xi(1), ...xi(n)|yi = y) = n\sum(k=1) k=1 P(xi(k)|yi = y)$$

Pseudo Code for Multinomial Naive Bayes

Step 1: Separate by class to calculate the probability of data by the class.

Step 2: The dataset needs two data sets from a given set of data.

Step 3: The data By Class require statistics from our training dataset.

Step 4: Gaussian Probability Density Function calculating the probability of observing a given value like x1 is difficult.

Step 5: Class Probabilities to calculate the probabilities for new data.

Decision Tree Algorithm

Decision Tree algorithm belongs to the family of supervised learning algorithms in which the goal of using Decision Trees is to introduce a training model that can use to predict the class or value of the target variable by learning simple decision rules concluded from prior data.Decision trees are a common learning method in machine learning. Good results have been achieved in classification, prediction and rule extraction. The tree structure includes three parts: a root node, a branch node, and a leaf node. It is also a decision node, usually representing a certain attribute of the sample to be classified in the data set. A branch is a different value of the root node, and a leaf node is a possible classification result. The Decision Tree algorithm divides the training set into relatively pure feature subsets and then recursively builds the Decision Tree. There are many algorithms based on Decision Trees. The most widely used is the C4.5 algorithm, which can process not only continuous and discrete attribute data but also data sets with missing values

Pseudo Code for Decision Tree

Step 1. Randomly select "x" values from total "y" values x<y

Step 2. Among x values find the node "a" using the best split point

Step 3. Split the node into two nodes using the best-split

Step 4. Frequently used 1-3 steps up to "b" no. of nodes have been reached

Step 5. Construct forest by frequently using 1-4 steps for "n" no. of times to generate no of trees.

Step 6: Return accuracy

$$E= - \sum p(i) \log2 p(i) \quad (2)$$

Statistical Analysis

SPSS software is used for statistical analysis of multinomial Naive Bayes and Decision Tree. Independent variables are e-Commerce sites like Flipkart, Amazon, Snapdeal, and Myntra. Dependent variables are e-Commerce sites and products. Independent T-test analysis is carried out to calculate accuracy for both methods.

2. Results

The proposed Multinomial Naive Bayes Algorithm and Decision Tree algorithm were run at different times in Jupiter with a sample size of 10. Table 1 represents the predicted accuracy of bug detection and alerting. These 10 data samples are used for each algorithm along with their loss values to calculate statistical values that can be used for comparison. From the results, it is observed that the mean accuracy of the Multinomial Naive Bayes Algorithm was 98.16% and the Decision Tree was 97.97%. Table 2 represents mean accuracy values for multinomial Naive Bayes and Decision Trees. The mean value of multinomial Naive Bayes is better when compared with the Decision Tree with a standard deviation of 98.1830 and 97.2120 respectively. It shows the Independent sample T-test data of multinomial Naive Bayes and Decision Tree with the significance value obtained is 0.429 (two-tailed, $p>0.05$). Figure 1 denotes the comparison of multinomial Naive Bayes and Decision Tree in terms of mean accuracy and loss.

Mean, Standard deviation and standard error mean for Multinomial Naive Bayes Algorithm are 98.1830, 0.84 and 0.26 respectively. Similarly, for Decision Tree, the mean, standard error mean and standard deviation are 1.38, 0.087 and 0.275 respectively. On the other hand, the loss values of multinomial Naive Bayes for mean, standard deviation and standard error deviation of 97.21, 0.929 and 0.294 respectively. For the Decision Tree, the loss values of the Decision Tree for mean, standard deviation and standard error mean are 2.55, 0.266 and 0.084 respectively.

The group statistics value along with mean, standard deviation and standard mean error for the two algorithms are also specified. The graphical representation of comparative analysis, mean of loss between two algorithms of multinomial Naive Bayes and Decision Tree are classified. This indicates that multinomial Naive Bayes are

significantly better with 98.16% accuracy when compared with Decision Tree classified accuracy of 97.97%.

3. Discussion

In the given review, the significance esteem acquired is 0.429 (Two-tailed, P>0.05) Which suggests that multinomial Naive Bayes seems, by all accounts, to be preferable over Decision Tree. Accuracy analysis of the Decision Tree classifier is investigated as 94% through the exactness of the Decision tree is 88%.

Comparative previous assessment of multinomial Naive Bayes over Decision Tree is depicted in this paper(Baye 2002). This clearly indicates that multinomial Naive Bayes appears to be a better classifier when compared to the Decision Tree classifier. This work shows comparative accuracy analysis between multinomial Naive Bayes and Decision Tree in which multinomial Naive Bayes Shows accuracy of 94% and Decision Tree shows an accuracy of 88%. In deep learning, the multinomial Naive Bayes is said to be a type of artificial neural network that detects and recognizes objects using the previously saved data sets. The connection between the two hidden layers is done by the Decision Tree. With this, the output layer can get data from the past and future states simultaneously. The accuracy of the Decision Tree Classifier is 86.2%. A Decision Tree is used for object identification, distance measurement, and other techniques. This application can be used on any device that uses a camera connected to computers, tablets and mobile phones. Few other applications are Flipkart, Amazon, Myntra, and Snapdeal. It is helpful in detecting a unique approach for detecting Novel e-bugs in e-Commerce sites.

The limitation of this study is that it takes a very long time to train a Naive Bayes algorithm, especially with large datasets. The future scope of this project is that the system should be explained to include a larger number of objects with lesser time consumption in training the data set.

4. Conclusion

In this project we have reviewed the technologies which are being used for finding and improving bug tracking systems. Further, we have introduced different techniques used to implement them. Present methods include database server, SQL and admin information. This comparison will help us in building our system more convenient and useful. From the research, we have proposed the system which will predict the time required for a particular task. Based on the analysis, multinomial Naive Bayes (98.16%) performs better than Decision Tree(97.97).

DECLARATIONS

Conflict of Interests

No irreconcilable situation in this original copy

Authors Contribution

Author SHK was involved in data collection, data analysis, and manuscript writing. Author SC was involved in the conceptualization, data validation, and critical review of a manuscript.

Acknowledgements

The creators might want to offer their thanks to Saveetha School of Engineering, and Saveetha Institute of Medical and Technical Sciences (Formerly known as Saveetha University) for giving the important framework to effectively do this work.

Funding

We thank the accompanying associations for offering monetary help that empowered us to finish the review.

1. Stigmata Techno Solutions.

2. Saveetha University.

3. Saveetha Institute of Medical and Technical Sciences.

4. Saveetha School of Engineering

5. References

Abbasi, Abdul Muqtadir. 2016. Categorization and Detection of Energy Bugs and Application Tail Energy Bugs in Smartphones.

Anupong, Wongchai, Lin Yi-Chia, Mukta Jagdish, Ravi Kumar, P. D. Selvam, R. Saravanakumar, and Dharmesh Dhabliya. 2022. "Hybrid Distributed Energy Sources Providing Climate Security to the Agriculture Environment and Enhancing the Yield." Sustainable Energy Technologies and Assessments. https://doi.org/10.1016/j.seta.2022.102142.

Balasubramanian, Kannan. n.d. "Threats and Attacks on E-Commerce Sites." Cryptographic Solutions for Secure Online Banking and Commerce. https://doi.org/10.4018/978-1-5225-0273-9.ch006.

Baye, Michael R. 2002. The Economics of the Internet and E-Commerce. Elsevier.

Bello, Paul J. 2011. The Bed Bug Combat Manual. AuthorHouse.

Bharathiraja, B., J. Jayamuthunagai, R. Sreejith, J. Iyyappan, and R. Praveenkumar. 2022.

"Techno Economic Analysis of Malic Acid Production Using Crude Glycerol Derived from Waste Cooking Oil." Bioresource Technology 351 (May): 126956.

Guo, Qi, Tianshi Chen, Haihua Shen, Yunji Chen, Yue Wu, and Weiwu Hu. 2011. "Empirical Design Bugs Prediction for Verification." 2011 Design, Automation & Test in Europe. https://doi.org/10.1109/date.2011.5763036.

Jothi, K. Jeeva, K. Jeeva Jothi, S. Balachandran, K. Mohanraj, N. Prakash, A. Subhasri, P. Santhana Gopala Krishnan, and K. Palanivelu. 2022. "Fabrications of Hybrid Polyurethane-Pd Doped ZrO2 Smart Carriers for Self-Healing High Corrosion Protective Coatings." Environmental Research. https://doi.org/10.1016/j.envres.2022.113095.

Kale, Vaibhav Namdev, J. Rajesh, T. Maiyalagan, Chang Woo Lee, and R. M. Gnanamuthu. 2022. "Fabrication of Ni–Mg–Ag Alloy Electrodeposited Material on the Aluminium Surface Using Anodizing Technique and Their Enhanced Corrosion Resistance for Engineering Application." Materials Chemistry and Physics. https://doi.org/10.1016/j.matchemphys.2022.125900.

Palanisamy, Rajkumar, Diwakar Karuppiah, Subadevi Rengapillai, Mozaffar Abdollahifar, Gnanamuthu Ramasamy, Fu-Ming Wang, Wei-Ren Liu, Kumar Ponnuchamy, Joongpyo Shim, and Sivakumar Marimuthu. 2022. "A Reign of Bio-Mass Derived Carbon with the Synergy of Energy Storage and Biomedical Applications." Journal of Energy Storage. https://doi.org/10.1016/j.est.2022.104422.

Pardo, Alejandro G. 2010. "BT-Engineered Bugs versus Insect Pests." Bioengineered Bugs. https://doi.org/10.4161/bbug.1.5.13656.

Ram, G. Dinesh, G. Dinesh Ram, S. Praveen Kumar, T. Yuvaraj, Thanikanti Sudhakar Babu, and Karthik Balasubramanian. 2022. "Simulation and Investigation of MEMS Bilayer Solar Energy Harvester for Smart Wireless Sensor Applications." Sustainable Energy Technologies and Assessments. https://doi.org/10.1016/j.seta.2022.102102.

Şerban, Camelia, and Andreea Vescan. 2021. "Towards a Neural Network Based Reliability Prediction Model via Bugs and Changes." Proceedings of the 16th International Conference on Software Technologies. https://doi.org/10.5220/0010600703020309.

Sumathy, B., Anand Kumar, D. Sungeetha, Arshad Hashmi, Ankur Saxena, Piyush Kumar Shukla, and Stephen Jeswinde Nuagah. 2022. "Machine Learning Technique to Detect and Classify Mental Illness on Social Media Using Lexicon-Based Recommender System." Computational Intelligence and Neuroscience 2022 (February): 5906797.

Thanigaivel, Sundaram, Sundaram Vickram, Nibedita Dey, Govindarajan Gulothungan, Ramasamy Subbaiya, Muthusamy Govarthanan, Natchimuthu Karmegam, and Woong Kim. 2022. "The Urge of Algal Biomass-Based Fuels for Environmental Sustainability against a Steady Tide of Biofuel Conflict Analysis: Is Third-Generation Algal Biorefinery a Boon?" Fuel. https://doi.org/10.1016/j.fuel.2022.123494.

Van Liew, Charles, Jessie M. Huisinga, and Daniel S. Peterson. 2021. "Evaluating the Contribution of Reactive Balance to Prediction of Fall Rates Cross-Sectionally and Longitudinally in Persons with Multiple Sclerosis." Gait & Posture 92 (November): 30–35.

Vickram, Sundaram, Karunakaran Rohini, Krishnan Anbarasu, Nibedita Dey, Palanivelu Jeyanthi, Sundaram Thanigaivel, Praveen Kumar Issac, and Jesu Arockiaraj. 2022. "Semenogelin, a Coagulum Macromolecule Monitoring Factor Involved in the First Step of Fertilization: A Prospective Review." International Journal of Biological Macromolecules 209 (Pt A): 951–62.

Yaashikaa, P. R., M. Keerthana Devi, and P. Senthil Kumar. 2022. "Algal Biofuels: Technological Perspective on Cultivation, Fuel Extraction and Engineering Genetic Pathway for Enhancing Productivity." Fuel. https://doi.org/10.1016/j.fuel.2022.123814.

Zimmermann, Thomas. 2009. Changes and Bugs: Mining and Predicting Software Development Activities. BoD – Books on Demand.

Gupta, Dharmendra Lal, and Kavita Saxena. "Software bug prediction using object-oriented metrics." Sādhanā (2017): 1-15..

M. M. Rosli, N. H. I. Teo, N. S. M. Yusop and N. S. Moham, "The Design of a Software Fault Prone Application Using Evolutionary Algorithm," IEEE Conference on Open Systems, 2011.

T. Gyimothy, R. Ferenc and I. Siket, "Empirical Validation of ObjectOriented Metrics on Open Source Software for Fault Prediction," IEEE Transactions On Software Engineering, 2005. Singh, Praman Deep, and Anuradha Chug. "Software defect

prediction analysis using machine learning algorithms." 7th International Conference on machine learning, Data Science & EngineeringConfluence, IEEE, 2017.

M. C. Prasad, L. Florence and A. Arya, "A Study on Software Metrics based Software Defect prediction using Data Mining and machine learning Techniques," International Journal of database Theory and Application, pp. 179-190, 2015.

Okutan, Ahmet, and Olcay Taner Yıldız. "Software defect prediction using Bayesian networks." Empirical Software Engineering 19.1 (2014): 154-181.

Bavisi, Shrey, Jash Mehta, and Lynette Lopes. "A Comparative Study of Different Data Mining Algorithms." International Journal of Current Engineering and Technology 4.5 (2014).

TABLES AND FIGURES

Table 1. Accuracy and Loss Analysis between Multinomial Naive Bayes and Decision Tree

S.No	GROUPS	ACCURACY	LOSS
1	Multinomial Naive Bayes	98.16	1.10
		97.00	1.09
		97.10	1.08
		97.21	1.23
		99.16	1.34
		99.14	1.87
		99.18	1.39
		98.21	1.57
		98.43	1.42
		98.24	1.74
2	Decision Tree	97.97	2.75
		96.12	2.34
		96.24	2.43
		98.32	2.23

		96.33	2.12
		96.28	2.90
		97.04	2.54
		97.32	2.87
		98.26	2.67
		98.24	2.66

Table 2:Group Statistical Analysis of NB and DT. Mean, Standard Deviation and Standard Error Mean are obtained for 10 samples. NB has higher mean accuracy and lower mean loss when compared to DT

	Algorithm	N	Mean	Std. deviation	Std. Error Mean
ACCURACY	multinomial naive bayes	10	98.18	0.84650	0.26769
	Decision Tree	10	97.21	0.92990	0.29406
LOSS	multinomial naive bayes	10	1.38	0.27568	0.08718
	Decision Tree	10	2.55	0.26610	0.08415

Table 3: Independent Sample T-test: CNN is insignificantly better than BRNN with p value 0.429 (Two tailed, p>0.05), the value of f is 0.65 and 0.04

		f	**sig**	**t**	**df**	**sig**	**Difference**	**std**	**Lower of 95.5% interval distance**	**upper**
Accuracy	Equal variance assumed	0.65	0.429	2.442	18	0.025	0.97100	0.397	0.13557	1.8064
	Equal Variance not assumed			2.442	17.843	0.025	0.97100	0.397	0.13504	1.8069

Loss	Equal variance assumed	0.04	0.952	-9.640	18	0.0	-1.16800	0.121	-1.42256	-.913
	equal variance not assumed			-9.640	17.97	0.0	-1.16800	0.121	-1.42258	-.913

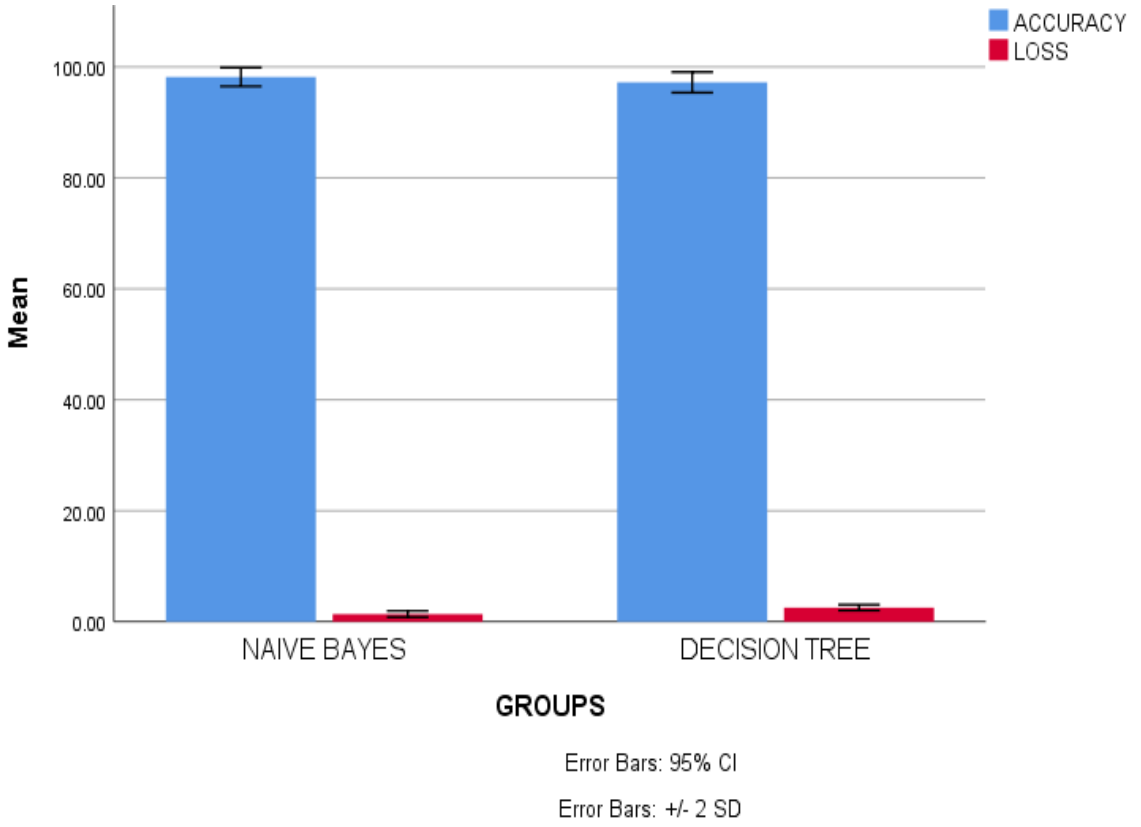

Fig 1. Comparison of NB and DT Classifier in terms of mean accuracy and loss. The mean accuracy of CNN is better than BRNN Classifier; Standard deviation of NB is slightly better than DT. X Axis: NB Vs DT Classifier and Y Axis: Mean accuracy of detection ± 2SD.