Original Article

Stacked Ensemble-IDS Using NSL-KDD Dataset

V.J. Immanuel Jeo Sherin[1], Dr.N. Radhika[2]
[1]Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.
E-mail: immanueljeosherin@gmail.com
[2]Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore India.
E-mail: n_radhika@cb.amrita.edu

Abstract

The intrusion detection system is a traffic monitoring unit that protects our network from numerous threats. It works as a monitoring unit with the ability to detect attacks in a real-time environment. Various techniques have been employed to make the IDS machine work with accuracy. To attain good accuracy, machine learning and deep learning are being utilized to train and evaluate the IDS machines. To prepare for the IDS's training and testing, before deploying them in real-time situations, a collection of real-world internet traffic records is stored with their traffic input features, this traffic records can be utilized to test the IDS machine against several attacks before they are employed into the real world. One such dataset is the University of New Brunswick's NSL-KDD dataset used to train and test the machine learning or deep learning models. In this paper, we will train and evaluate our model using the proposed stacked ensembled machine learning model, and with the NSL-KDD dataset, compare it based on various evaluation metrics against standard machine learning methods and some earlier proposed research.

Keywords: Machine Learning Model, Ensembled Models, NSL-KDD Dataset, Stacked Ensembled Models.

DOI: 10.47750/pnr.2022.13.S03.057

INTRODUCTION

The Intrusion detection system is used in the host to detect if any malicious activities are performed in real time networks. It could be based on rules or signatures. When using a rule-based approach, the rules will be designed to check for conditions, and if they are met, an alert will be issued. In the same way for signature-based intrusion detection systems, signatures for malicious payloads are stored in the intrusion detection system database and will be verified to check if the signature matches with the one in the database, if it is verified, alert will be given.

Machine learning is a subset of artificial intelligence that enables software programs to enhance their prediction ability without being explicitly programmed to do so. Machine learning algorithms anticipate new output values by using existing data as input.

Ensemble techniques are a type of machine learning methodology in which many base models are combined into a single best-fit prediction model. There are various techniques that can be utilized to implement ensembled model such as stacking, boosting, and bagging. In Bagging, the model often analyses homogenous weak learners, learns them in parallel, and then combines them using a deterministic averaging strategy. In boosting, the model generally considers identical weak classifier, learns them sequentially in a relatively self-correcting manner, employs a base model that is dependent on the prior ones, and then combines them in a deterministic manner. The model in Stacking analyses a range of weak learners, learns them in simultaneously, and then combines them by training a meta-model to generate a forecast based on the predictions of the weak models.

The goal of this research is to employ a stacked ensembled machine learning model to distinguish between four types of attacks: DOS, Probe, U2R, and R2L. The proposed stacked machine learning model is developed using random forest classifier ranking and forward selection to pick features. The selected features are trained on the training dataset, then they are tested on the testing dataset. It is then compared to a variety of traditional classification techniques as well as some earlier NSL-KDD Dataset based models.

The following section of the paper will be as follows. The literature review is depicted in Section II. The proposed methodology and strategies are briefly described in Section III. The implementation of the real dataset is shown in Section IV, along with preliminary results. Finally, Section V discusses the conclusion and future research.

LITERATURE REVIEW

Different machine learning and deep learning classification techniques have been presented in recent years to evaluate the IDS machine using the NSL-KDD dataset given. This

section provides an overview of the many strategies used to evaluate the NSL-KDD dataset, as described in the following articles.

Multiple articles have evaluated the IDS model using the NSL-KDD dataset using different classification methods such as binary and multiclass classification. The dataset is divided into normal and attack packets, and the IDS is trained to identify whether a packet is normal or attack using binary classification. In multiclass classification, the dataset is separated into four primary attacks and the IDS is trained to detect the attack type.

Lukman, Hakim in [2] discussed gain ratio and relief are some of the feature selection strategies covered for feature selection and tested the features on various machine learning classification algorithms. The research concluded with the conclusion that feature selection was able to boost the IDS's performance but there was a slight reduction in the accuracy score.

In [1] Rahman and Mashuqur discussed a hybrid machine learning model that uses 17 features identified by the chi-squared test and uses K-Means as an unsupervised learning technique, the light gradient boosting machine was used, and as a supervised approach, the Light Gradient Boosting Machine was used. The 17 features chosen from the CHI square test will be used for testing and training. The model was put to the test for binary classification of attack or regular packets and scored 90.4 percent accuracy.

In [3] Mukherjee proposed a new feature selection model that can be used to select features by doing test and train with naïve bayes classifier.23 features were selected based on the feature selection method and this was trained and tested with the naive bayes classifier. In [4] Tang, Tuan A selected 5 features to train this model, the author uses Deep learning models to educate and test, the two models Deep neural network and Gated Recurrent neural network were used, In this DNN got an accuracy of 80.7% and G-RNN got an accuracy of 89% for multiclass classification.

PROPOSED METHODOLOGY

This section gives a brief overview of the proposed technique for evaluating the model: Ensembled Machine Learning (Section III - A), Stacking Ensembled Model (Section III - B), Random Forest (Section III - C), and KNN (Section III - D), as well as the methodology outlined in (Section III - E).

A. Ensembled Machine Learning

Ensemble machine learning approaches incorporate inputs from several learning models to help make more accurate and better judgements. Noise, variation, and bias are the main drivers of error in machine learning models. Ensemble approaches in machine learning help to reduce these error-causing elements, ensuring that machine learning algorithms are accurate and stable. Bagging, boosting, and stacking models are techniques among the ensembled models. The technique of fitting numerous decision trees to various samples of the same dataset and then averaging the results is known as bagging. Stacking is the process of fitting numerous types of models to the same data and then using another model to identify the optimal approach to integrate the results. Boosting entails adding ensemble members in a sequential order to correct past model predictions and obtain a weighted average of the forecasts.

B. Stacked Ensembled Machine Learning

Stacking is an ensemble modelling strategy that entails using data from many models' predictions as features to create a new model and make predictions. Base models are the models that are integrated, while meta-features are the predictions that are used as additional features to train the final model. A stacking model's architecture consists of two or more base models (also known as level-0 models) and a meta-model (also known as a level-1 model) that incorporates the base models' predictions. The meta-model is given training utilizing data predictions based on out-of-sample data given Through means of basic models non-training data is fed into the base classifiers, predictions are produced, and the predicted outputs, combined with the expected predictions, include the input and output combinations of the training data used to construct the meta-model.

C. Random Forest

The random forest algorithm is a bagging modification. To construct a statistically independent forest of decision trees, bagging and feature randomness are used. Feature randomization produces a random collection of features, resulting in minimal correlation in decision trees. The primary distinction between decision trees and random forests is that choice trees analyse all possible feature splits, whereas random forests only consider a subset of them. Before training, the three fundamental hyperparameters of random forest algorithms must be determined. Node size, tree count, and number of features sampled are examples of these variables. The random forest classifier may then be used to solve regression or classification issues.

D. KNN

K-nearest neighbours is a supervised machine learning algorithm that may be used to solve classification and regression prediction issues. However, it is usually used to address classification and prediction issues. KNN is a lazy learning algorithm since it lacks a dedicated training phase and instead uses all of the input for training and classification. KNN is a non-parametric learning method since it makes no assumptions regarding data. The K-nearest neighbours method predicts new data points based on feature similarity, which means that the new data point will be assigned a value depending on how similar it is to the points

in the training data set.

E. Methodology

The following architecture is proposed in this research to successfully analyze the IDS utilizing stacked ensembled machine learning techniques. To create a training model, the proposed stacked model is applied to the data used for training. The test data is used to create a predictive model. The output of the predictive model is compared to the model created using trained data.

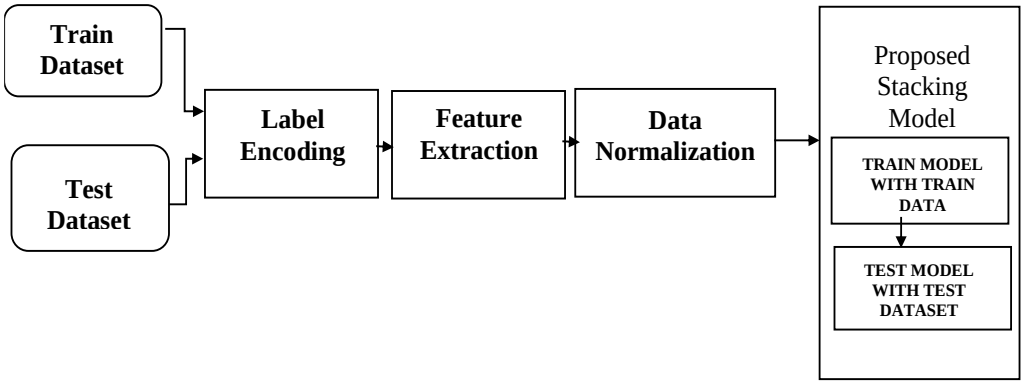

Fig. 1; Proposed Methodology

CASE STUDY

This section contains observational research in which the suggested stacking model is used to the training and testing model to evaluate the IDS, with random forest, KNN as base models and random forest as final estimator.

A. Dataset

The NSL-KDD dataset contains internet traffic logs and is an enhanced version of the KDD-99 dataset. The University of New Brunswick introduced the dataset. The dataset comprises 43 characteristics, with 41 representing the traffic input, a label for the traffic record indicating whether it is an attack or a typical packet, and a score for the intensity of the traffic input in the record. DDOS, Probe, U2R, and R2L are the four key attacks in the NSL-KDD dataset.

DDOS: DDOS is a type of assault that aims to disrupt legitimate users access to services. Flooding is a common method of attack that includes sending a large number of packets to the target machine.

Probe: Probe is a type of attack in which the attacker creates a packet and sends it to the target system in order to obtain information.

U2R: In a U2R attack, the attacker starts at the user level and then attempts to get root level access by exploiting the target's vulnerabilities.

R2L: The R2L attack attempts to gain local access to a machine that is located in a remote location.

The key attacks in the dataset are listed above and each attack is divided into several types in the dataset which are grouped into the major four.

Table 1. Subclass Information of Attacks in Dataset

Class	No of subclasses in the dataset	Example Attacks in the class
DDOS	11	Apach2, Neptune, Teardrop
Probe	6	Ipsweep, Nscan, Satan
U2R	7	Perl, Ps, Rootkit
R2L	15	Imap, MultiHop, SendMail

Dimensions of the Training set: (125973, 43), The number of rows in the training set is 125973, while the number of columns in the training set is 43. **Dimensions of the Testing set: (22544, 43)**, The number of rows in the training set is 22544, while the number of columns in the training set is 43.

B. Evaluation Metrics

The model will be evaluated based on the metrices such as accuracy, precision, recall.

- Accuracy is defined as the measure of correctly predicted observations with total observations.

Accuracy = Number of Correct Prediction / Total Number of Correct Prediction (1)

- Precision is defined as the percentage of accurately anticipated positive observations that occur in the total number of positive observations.

Precision = True Positive / True Positive + False Positive (2)

- Recall is the ratio of expected positive observations to actual positive observations in a class.

Recall = True Positive/True Positive + False Negative (3)

- The F1 score is calculated using the harmonic mean of accuracy and recall.

*F1 Score = 2 * Precision * Recall / Precision + Recall* (4)

C. Label Encoding

It is an important pre-processing step for the structured dataset in supervised learning. Label encoding is a common encoding method for categorical information. In this system, each label is assigned a unique integer based on alphabetical order. The label encoding Technique is used to assign a unique integer to the columns service, flag, label, and protocol type in the dataset.

D. Feature Extraction

The features are critical to any model's efficiency during training and testing. The dataset contains 41 input features. however, training the model with all 41 input features did not result in significant prediction output, therefore choosing the key characteristics to train the model was one of the most crucial elements of improving the IDS model's performance.

Random Forests are frequently used in data science processes for feature selection. This is because the tree-based techniques utilised in random forests are organically prioritised depending on how effectively they promote node integrity. This is the overall impurity reduction across all trees. The nodes with the largest drop in impurity are at the top of the trees, while those with the least decrease are at the bottom. We may acquire a subset of the most significant properties by cutting trees below a specific node. Random forest uses gini significance or mean reduction in impurity to compute the value of each feature. Gini significance is the total decrease in node impurity. This is how much the model fit or accuracy worsens when a variable is removed. The bigger the significance of the variable, the larger the decline. In this scenario, the mean decrease is a crucial metric for variable selection. The Gini index can be used to indicate the entire explanatory power of the variables. After training with a random forest classifier for feature extraction, each input is delivered.

Table 2. Random Forest Classifier Feature Scores

Feature	***Score***
Src_bytes	1.809939e-01
Dst_bytes	1.220847e-01
Flag	9.128644e-02
Dst_host_same_srv_rate	8.849899e-02
Same_Srv_rate	6.138564e-02
Diff_srv_rate	6.046698e-02
Dst_host_srv_Count	4.110554e-02
Protocol_type	3.996323e-02
Dst_host_diff_srv_rate	3.972354e-02
Count	3.559104e-02
Service	3.263855e-02
Dst_home_same_Src_port_rate	2.848263e-02
Dst_host_count	1.785421e-02
Logged_in	1.778762e-02
Dst_host_serror_rate	1.700376e-02
Srv_count	1.700238e-02
Serror_rate	1.690019e-02
Dst_host_srv_diff_host_rate	1.424102e-02
Srv_serror_Rate	1.290367e-02
Dst_host_rerror_rate	1.019157e-02
Dst_host_Srv_serror_Rate	9.832681e-03
Hot	9.326493e-03
Dst_host_srv_error_rate	8.844310e-03
Num_compromised	6.388590e-03
Wrong_fragment	4.367317e-03
Duration	3.915526e-03
Srv_diff_host_rate	3.625837e-03
Srv_rerror_Rate	3.292882e-03
Rerror_rate	2.384324e-03
Is_guest_login	9.415394e-04
Num_root	3.118220e-04
Num_failed_logins	2.309168e-04
Num_file_creations	1.426215e-04
Root_shell	9.933863e-05
Num_access_files	5.437649e-05
Num_shells	5.183076e-05
Land	3.451448e-05
Urgent	2.821069e-05
Su_attempted	2.089862e-05
Is_host_login	3.061455e-07
Num_outbound_cmds	0.000000e+00

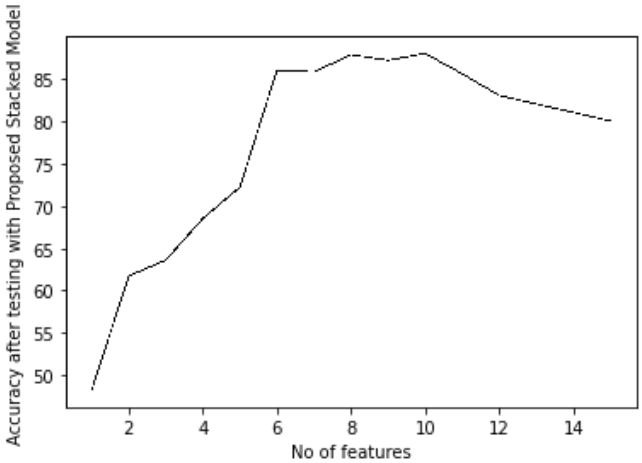

Fig. 2; No of Features VS Accuracy Score

The variables are picked using a forward selection strategy, and the features are ordered from highest to lowest based on the score. Forward selection is a type of stepwise regression in which variables are gradually added to a blank model. You add the one variable that improves your model the most each

time you advance a step. As can be seen in the graph above, as forward selection progresses, the accuracy decreases as th e number of features increases. We employ the service input feature because it is categorical, and the rest of the categorical features are already covered in the 8 selected features, as indicated in the table below. We select 9 features after forward selection, which are the top 8 ranks from the random forest classifier result and a categorical variable.

Table 3. Selected Features

	Features
1	Src_bytes
2	Dst_bytes
3	Flag
4	Dst_host_same_srv_rate
5	Same_Srv_rate
6	Diff_srv_rate
7	Dst_host_srv_Count
8	Protocol_type
9	Service

E. Data Normalization

The data is rescaled between 0 and 1 using the Min-Max normalization technique. The algorithm works by taking each value and subtracting the minimal value from it. X stands for the absolute minimum (X). After that, the data must be adjusted to be on the upper bound 1, which may be done by dividing each value in the data by its original range. The NSL-KDD dataset after feature extraction is normalized using the min-max normalization method.

F. Training and Testing the Model

Random forest and KNN are the foundation models, and random forest is the meta model or final estimator model. Experimental trials were used to choose the base and final estimator models. The training dataset is used to train and the testing dataset is used to test the basic model. The result will be predicted by combining the predictions from the base model by the meta model.

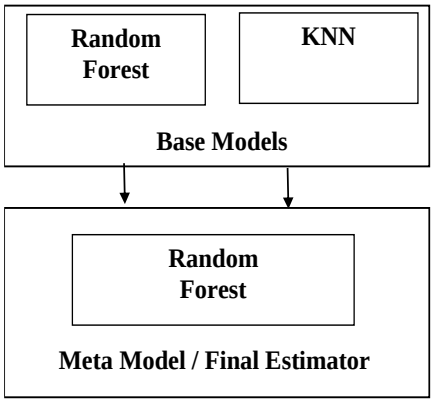

Fig. 3; Proposed Stacked Ensemble Model

RESULTS AND DISCUSSIONS

Performance indicators such as accuracy, precision, recall, and FI Score are used to assess the model's efficiency. The proposed model's result is shown in the table below.

Table 4. Model Performance

Class	***Precision***	***Recall***	***F1 Score***
DDOS	96%	95%	96%
Probe	73%	92%	81%
U2R	98%	76%	85%
R2L	48%	45%	47%

The proposed model has an overall accuracy of **89.98** percent and just 9 features are used to train it.

The models are compared with some of the proposed IDS models and standard classification models with the selected features.

Table 5. Comparison of Proposed Model with Different works and Standard Models with features

Model	**No. of features used**	**Classification**	**Accuracy**
Hybrid Machine Learning Model proposed by [1] Rahman	17	Binary	90.4%
Deep Learning – DNN [4]	5	Multiclass	80%
Deep Learning – GNU RNN [4]	5	Multiclass	89%
Naïve Bayes [3]	23	Multiclass	97%
Naïve Bayes	9 (Proposed Features)	Multiclass	71%
Logistic Regression	9 (Proposed Features)	Multiclass	73%
Random Forest	9 (Proposed Features)	Multiclass	85%
Proposed Stacked Model	9 (Proposed Features)	Multiclass	89.98%

The below table explains the results with all 41 features against the standard classification algorithm and the proposed stacking model for multiclass classification.

Table 6. Comparison of Proposed Model with 41 Features and Standard Models with 41 features

Model	No. of Features Used	Classification	Accuracy
Naïve Bayes	41	Multiclass	52.09%
Random Forest	41	Multiclass	81.1%
Logistic Regression	41	Multiclass	52.52%
Proposed Stacked Model	41	Multiclass	81.67%

The proposed stacking model performs better than the deep learning models DNN and GNU-RNN [4] based on the above results. It also works equivalent to the binary classification hybrid machine learning model proposed by Rahman [1] with 17 features. The model clearly outperforms the traditional classification algorithm when only the selected features are used and with the 41 features.

CONCLUSION

The stacking model appears to be a good way to develop the IDS, and it can be improved by stacking it with deep learning models. The random forest classifier feature extraction technique clearly demonstrates that employing the selected 9 features rather than all 41 features is preferable. Instead of utilizing the feature extraction model to obtain features for all attacks, focusing on employing attack-specific characteristics to train the model could minimize the model's error prediction rate. This are some of the ideas of future research with the IDS.

REFERENCES

Rahman, Mashuqur & Kamruzzaman, Niton & Akter, Nasrin & Arbe, Nafija & Rahman, Md. Mahbubur. (2021). Network Intrusion Detection Using Hybrid Machine Learning Model, 1-8. 10.1109/ICAECT49130.2021.9392483.

L. Hakim, R. Fatma and Novriandi, "Influence Analysis of Feature Selection to Network Intrusion Detection System Performance Using NSL-KDD Dataset," 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), 2019, pp. 217-220. doi: 10.1109/ICOMITEE.2019.8920961.

Mukherjee, Saurabh & Sharma, Neelam. (2012). Intrusion Detection using Naive Bayes Classifier with Feature Reduction. Procedia Technology, 4. 119–128. 10.1016/j.protcy.2012.05.017.

Tang, Tuan A., Lotfi Mhamdi, Des McLernon, Syed A.R. Zaidi, Mounir Ghogho, and Fadi El Moussa. 2020. "DeepIDS: Deep Learning Approach for Intrusion Detection in Software Defined Networking" Electronics 9, no. 9: 1533. https://doi.org/10.3390/electronics9091533

N. Kunhare and R. Tiwari, "Study of the Attributes using Four Class Labels on KDD99 and NSL-KDD Datasets with Machine Learning Techniques," 2018 8th International Conference on Communication Systems and Network Technologies (CSNT), 2018, pp. 127-131, doi: 10.1109/CSNT.2018.8820244.

R. Alzahrani, Abdulsalam & Alenazi, Mohammed. (2021). Designing a Network Intrusion Detection System Based on Machine Learning for Software Defined Networks. Future Internet. 13. 111. 10.3390/fi13050111.

Vipin, Das & Vijaya, Pathak & Sattvik, Sharma & Sreevathsan, & MVVNS.Srikanth, & T, Gireesh. (2010). Network Intrusion Detection System Based on Machine Learning Algorithms. International Journal of Computer Science & Information Technology. 2. 10.5121/ijcsit.2010.2613.

B. Pavlyshenko, "Using Stacking Approaches for Machine Learning Models," 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), 2018, pp. 255-258. doi: 10.1109/DSMP.2018.8478522.

Kunal and M. Dua, "Machine Learning Approach to IDS: A Comprehensive Review," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 117-121. doi: 10.1109/ICECA.2019.8822120.

S. Sriram, R. Vinayakumar, M. Alazab and S. KP, "Network Flow based IoT Botnet Attack Detection using Deep Learning," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020, pp. 189-194, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162668.

S. Jayaprakash and K. Kandasamy, "Database Intrusion Detection System Using Octraplet and Machine Learning," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 1413-1416, doi: 10.1109/ICICCT.2018.8473029.

V. Borhade, A. Nayak and R. Dakshayani, "Intrusion detection: A machine learning approach" in Algorithms for Intelligent Systems, Singapore: Springer Singapore, pp. 555-561, 2020.

X. Gao, C. Shan, C. Hu, Z. Niu and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection", IEEE Access, vol. 7, pp. 82512-82521, 2019.

P. Louridas and C. Ebert, "Machine learning", IEEE Softw., vol. 33, pp. 110-115, 2016.

M. Rm and D. Radha, "A Comprehensive Approach for Network Security," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 420-426. doi: 10.1109/ICICCT.2018.8472952.

S. Arvind and V. A. Narayanan, "An Overview of Security in CoAP: Attack and Analysis," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 655-660, doi: 10.1109/ICACCS.2019.8728533.

V. Sidharth and C. R. Kavitha, "Network Intrusion Detection System Using Stacking and Boosting Ensemble Methods," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021, pp. 357-363, doi: 10.1109/ICIRCA51532.2021.9545022.

H. Nkiama, S. Z. M. Said, and M. Saidu, ‘‘A subset feature elimination mechanism for intrusion detection system,’’ Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 3, pp. 148–157, 2016.

C. Chio and D. Freeman, Machine Learning and Security: Protecting Systems with Data and Algorithms. Newton, MA, USA: O’Reilly Media, 2018.

Ibrahim, S., & Koksal, M. E. (2021). Realization of a fourth-order linear time-varying differential system with nonzero initial conditions by cascaded two second-order commutative pairs. Circuits, Systems, and Signal Processing, 40(6), 3107-3123.