Course	Author	Date	Revision
ICAT3210, Applied Machine Learning, Autumn 2021	Petri Välisuo	30.12.2021	A

Evaluation of course project

Here is the evaluation of the course project:

Title	Network Intrusion Detection System based on NSL-KDD Dataset
Pages	32, 20 effective
Authors	Hoang Nguyen Duc
Course	ICAT3210, Applied Machine Learning, Autumn 2021
Grade	55 / 60

1 Evaluation

1.1 Background

The problem selected for this work, is to classify network traffic into attacks of four classes:

1. Denial of Service (DoS)
2. Probe to find out system's details
3. Remote to Local (R2L) and
4. User to Root (U2R) attack

The topic is as relevant as always since the beginning of Internet, or perhaps even more so when the critical infrastructures are controlled by IoT devices and cyber-attacks have became tactical tools for governmental bodies as part of foreign intelligence and warfare.

The data set used is often used in NIDS literature and is suitable to the purpose.

Some relevant literature is cited in the introduction and the topic is clearly explained and put in context.

1.2 Materials and methods

1.2.1 Data

The data used was the NSL-KDD data set. It contains separate test set and training set and 40 features from each sample:

Data	*N*	*P*
NSL-KDD training set	148,517	40
NSL-KDD test set		40

1.2.2 Preprocessing and further feature selection

The statistics of the data set was analyzed, and it did not contain missing values. One feature seemed unnecessary and was removed.

The intrusion types were placed in the four categories mentioned above. This results in an unbalanced data set having 53 385, 14 077, 3 749, 252 and 7 705 in classes DoS, Probe, R2L, U2R and normal respectively, having 79 168 samples total.

Categorical features were encoded using one-hot encoding and all features were normalized to zero mean and unit variance to be suitable for most machine learning algorithms. Decision tree based methods could have been applied also to categorical data, without normalization. After one-hot encoding, the number of features is 33+88 = 121.

By the way, normalizing data to unit variance (or unit std) does not mean that all values are in the range [0,1].

The dimensionality is then reduced to by selecting 22 most important features of PCA transformed data, which contains 95% of the variance.

The data set was randomly splitted to train and test sets having 75% and 25% of samples respectively. For unbalanced data, this splitting would be best to make each output class separately or at least check that also the smallest class have enough samples in test and training sets.

1.2.3 Model design and rationale

Three different models were trained: KNN, SVM and MLP.

The KNN was trained using the neighborhood size as 5 by exhaustively testing several values.

The SVM model is trained in a similar way than the KNN model. It is not explained how the hyper-parameters, third order polynomial kernel was selected.

The rationale for using MLP classifier is to compare it with KNN and SVN and the author also believes that it could be faster. When the PCA has been fit to the data, the transformation of the new data requires only few (22 in this case) vector products, and it should not be very slow. NN classifier can become slow if the training data set is large, but SVM should not be slow. My understanding is that the PCA-SVM prediction should not be much slower than MLP with two hidden layers, but it would be interesting to actually compare the prediction speeds. Training time should not be a problem, since training can be done off-line.

The MLP model contains two hidden layers having 64 and 32 perceptrons respectively, and the total model contains 10 053 trainable parameters. It is not explained why exactly this kind of model was constructed, but they seldom are. The amount of training data may easily become an obstacle for training complex MLP models, but it seems to be sufficient in this case.

1.2.4 Model validation

The models were validated using 5-fold cross validation and studying the accuracy with few precision metrics, a confusion matrix and a ROC curve.

In this kind of case, it is important to study the performance in each category in addition to the overall performance. In this work, this is done very well using confusion matrix and plotting the ROC-curves separately for each class.

ROC curves are not drawn for the MLP, but according to the confusion matrix and accuracy metrics, the model works the best.

1.3 Results

1.3.1 Quality of results

The overall accuracy of all models were very high but there are issues in predicting the class with the least amount of training samples. Data augmentation strategies could have been tried to make the data set more balanced, but it has also some disadvantages. One interesting method for handling unbalanced data is Synthetic Minority Oversampling Technique (SMOTE) as explained in “SMOTE for Imbalanced Classification with Python”.

For intrusion detection, it may be desirable to tune the classifier so that the sensitivity would be high in all classses at the cost of precision. The ROC curve is a good method to analyze the capabilities of the classifier and selecting the decision threshold so that an optimal balance between costs related to false alarms and undetected intrusion is found.

1.4 Discussion and conclusion

1.4.1 Intrepretation of the results

The performance of the selected machine learning methods are compared with each other and the applicability and conditions for using machine learning for the purpose is discussed.

1.4.2 Comparison with previous results

The results were not compared with previous results from the literature. This is somewhat compensated by testing several models.

1.4.3 Critical evalution of own work

The authors have critically evaluated the performance of the models, and strived for improving the results using several models. The accuracy and applicability of the prediction results for practical purposes has not been analyzed.

Course	Author	Date	Revision
ICAT3210, Applied Machine Learning, Autumn 2021	Petri Välisuo	30.12.2021	A

2 Conclusion

The work is documented very well with the references and the selection of the data set is good. The testing of several methods is a benefit. The full source code has been included as appendices with nice syntax highlight.