CIS 6650: FUNDAMENTALS OF COMPUTER SECURITY

DEEP LEARNING FOR INTRUSION DETECTION SYSTEMS (IDS)

December 20, 2018

John Beninger
Omar Irfan Khan
University of Guelph
School of Computer Science

INTRODUCTION

A number of tools exist to provide multi-layered computer security, one of which is an intrusion detection system (IDS). An intrusion detection system monitors network traffic for malicious activities and sends an alert upon detection. However, it is a passive device so it operates parallel to the network as compared to an intrusion prevention system (IPS) [6]. An intelligent IDS combines either machine learning algorithms, fuzzy classifiers or a hybrid approach which merges several algorithms to effectively detect and alert the user [2]. Moreover, the need for intelligent IDS is pronounced because existing intrusion detection systems have high rates of both false positives and false negatives [1], [24].

A variety of machine learning methodologies have been applied to make IDS more robust and intelligent. Nevertheless, most machine learning algorithms are classified shallow learning which means that they need handcrafted features for each target problem. They have trouble dealing with dynamic datasets because multiple classification tasks will result in decreased accuracy. Conversely, deep learning is well suited for IDS as it needs a large amount of data and it can learn features directly for each target problem from the dynamic dataset [25].

We present an evaluation of the relative performance of a deep learning classifier against existing IDS classification algorithms on a high quality data set. Further, literature review has indicated deep learning performance is highly dependent on the type of dimensionality reduction applied. For example [13] found that a deep learning model using self-taught learning via autoencoder (a dimensionality reduction technique) outperformed other deep classifiers by a large margin. Work by [20] has also shown better performance of multi-layer neural network systems than [13] by using a random forest regressor for feature selection. Therefore we also present an experimental framework, and accompanying code, for mixed factorial analysis on the effect of dimensionality reduction technique and classification algorithm on measures of classification success. The primary contribution of this work is the rigorous comparison of several classification algorithms on a high quality dataset that is relevant to the Canadian context [20]. Further, rather than acting as binary classifiers, the algorithms applied in this work have been implemented to predict the single most likely classification from a set of attack vectors. While this incurs the drawback of reduced classifier precision, sensitivity, and f-measure it means these systems always present a single class label: that of the most likely attack. To our knowledge this is the first implementation of a general, non-binary classifier on this data set. This work also contributes the tools to evaluate dimensional reduction options for future IDS classification studies.

BACKGROUND

Support vector machines (SVM) [11],[19], Self organizing maps(SOM) [3], Random forest (RF) [7],[22] , K nearest neighbor(KNN) [14], Artificial neural networks (ANN) [15] and several other techniques [2] ,[23] have been used as classifiers in IDS. Evolutionary algorithms (EA) such as genetic algorithms have also been used in combination with IDS either on their own or paired with an SVM [18]. Using genetic algorithms alone results in sub-par performance and gives a rise to higher false positive rates [9].

In recent years deep learning has gained popularity due to increases in computational power and availability of data [12]. In [10] the authors propose a deep learning approach with deep belief networks (DBN) paired with SVM. The results show that deep learning methods work very well in intrusion detection systems as compared to previous methods. Self taught learning (STL) and soft max regression (SMR) were evaluated against each other on different classes of attack data. Furthermore, in [17] deep neural networks (DNNs) have also been trained using the a well known, though limited, data set known as NSL-KDD. In this case, the authors fine-tuned their DNNs by adding two auto encoders to learn from unlabeled data faster which improves the performance of DNNs [8],[4]. In addition, the DNN was successful in identifying cases involving multiple attack vectors.

Recurrent neural networks (RNNs) have also been implemented in IDS'. Mansour Sheikhan et al. propose a three layer RNN [21] however, their RNN was not able to learn high dimensional features. In anther recent study, the authors [25] propose a RNN-IDS model which evaluates the models performance against other machine learning methods such as J48, ANN, RF, SVM and more. The authors found that the proposed model had a higher accuracy and lower false positive rate than traditional machine learning methods. A recent survey of deep learning methods [13] compares DNNs, RNNs based long short term memory (LSTM) and STL based on autoencoders. The authors found that autoencoders were able to classify attack types with higher accuracy than other methods. A cleaned version of NSL-KDD was used which removes duplicate entries in the dataset.

Most of these publications rely on variations of the KDD data set: either the KDDCUP or NSL-KDD data set. Only one paper used a relatively new data set which is the ISCX 2012 data set (also known as: ISCXIDS2012). Most old data sets, including KDD, have a lot of duplicate data, lack attack profiles and traffic diversity, and contain a low volume of data for each attack. The UNSW-NB 15 [16] data set is a recent data set but it does not have a large volume of data as compared to CICIDS2017 and UNSW-NB15.

Prior comparative evaluation of the deep learning methods in both the KDD and NSL-

KDD data sets [13] a self-taught learning (STL) deep learning approach will be implemented using an autoencoder for feature selection. This is because this method was found to outperform both multi-layer perceptron (MLP) and recurrent neural network (RNN) approaches in both precision and recall [13].

The performance of both deep and shallow neural network classifiers on IDS classification problems is highly dependent on the type of dimensional reduction applied to data sets in the work evaluated. The best classification achieved in [13] used an autoencoder to perform an initial compression of the data prior to classification. Further by using a random forest regressor for feature selection [20] were able to achieve better multi-layer perceptron performance than all deep learning classifiers used by [13] except for the autoencoder. While [13] compared many of the leading machine learning algorithms using a high quality data set from University of New Brunswick, it appears their classifications were binary. The limitation to this approach is that a suite of binary classifiers may identify a single event as being several different attacks at the same time. As such, to our knowledge, the following gaps have been identified in the literature: There is no rigorous comparison of leading deep learning IDS techniques on a recent, high quality, data set relevant to the Canadian context. There is no rigorous comparison of optimal dimensional reduction techniques for deep learning IDS classification on recent data. There is no non-binary general classification comparison of machine learning classifiers published for the most recent ICISSP data set, (though there is an excellent binary comparison [20].

METHODS

0.1 DATA PREPARATION

Because of the issues with other existing datasets outlined by Sharafaldin Lashkari Ghorbani [20] and the degree to which security threats are spatially and temporally specific, our work tests intrusion detection classifiers specifically on the CICIDS2017 data set [20]. Previous work has evaluated the effectiveness of KNN, RF,Iterative Dichotomiser 3 (ID3), Adaptive boosting (Adaboost), Naive-Bayes, Quadratic Discriminant Analysis (QDA), and multi-layer perceptron algorithms in terms of precision, recall, F-measure and execution time on this dataset [20]. Of these algorithms, KNN, RF, and ID3 achieved the highest F-measure scores, demonstrating both high precision and high sensitivity [20].

In addition to providing performance measures, the CICIDS2017 dataset also lists recommendations for feature selection by attack type. This project compares the performance

of a modern deep learning method against a random forest (RF) classifier and K-nearest Neighbours (KNN). In order to ensure fair evaluation we implemented these algorithms on the subset of CICIDS2017 containing DoS GoldenEye, SSHPatator, FTPPatator, Bot, PortScan, DDos, and benign activity patterns. Due to the small number of data points representing other attack vectors, and the data hungry nature of deep learning algorithms, the above were the only attack vectors for which the data set was expected to reliably support deep learning. Feature selection was based on the suggested features published for these conditions in [20] and all of the algorithms were provided data with the same 19 features. Code is also provided to perform a dimensional reduction on CICIDS2017 to a chosen number of features by compression using a deep autoencoder. The application of our deep learning classifier to the output from this compression replicates the self-taught learning approach identified as optimal by [13].

Data points were selected from the CCIDS2017 data set by fully sampling points corresponding to DoS GoldenEye, SSHPatator, FTPPatator, and then randomly under-sampling points corresponding to DDoS, portscan, and benign. This was done to reduce the sharp over-representation of the later three attack vectors in the data set, in order to prevent over-training to only those labels. All points were randomly shuffled. In order to prevent testing on data used for tuning, 10% of the CICIDS2017 data points were set aside for tuning hyper-parameters which affect the way in which each classifier trained. The remaining 90% of selected data was provided to the algorithms under evaluation for training and testing.

0.2 TUNING AND HYPER PARAMETERS

RF, KNN, multinomial naive bayes and QDA were chosen since they are the most commonly used algorithms and were also used in the UNB paper as a comparison. The dataset was split 90/10 where 90% was for training and testing. The 10% contributed to fine tuning the hyper parameters for each algorithm. There are three main features to tune in a random forest, which are the max_features,the n_estimators and criterion. n_estimators indicates the number of trees in the forest while, max_features determine the number of features for the best split. Criterion measures the quality of a split. There are mainly two ways to choose criterion either by gini (which looks for impurity) or by entropy (which looks for information gain). The n_estimators were varied from 10-200 to determine which n_estimator gave the best accuracy.

The most consistent accuracy obtained is from n_estimators=100 shown in Figure 1 below.

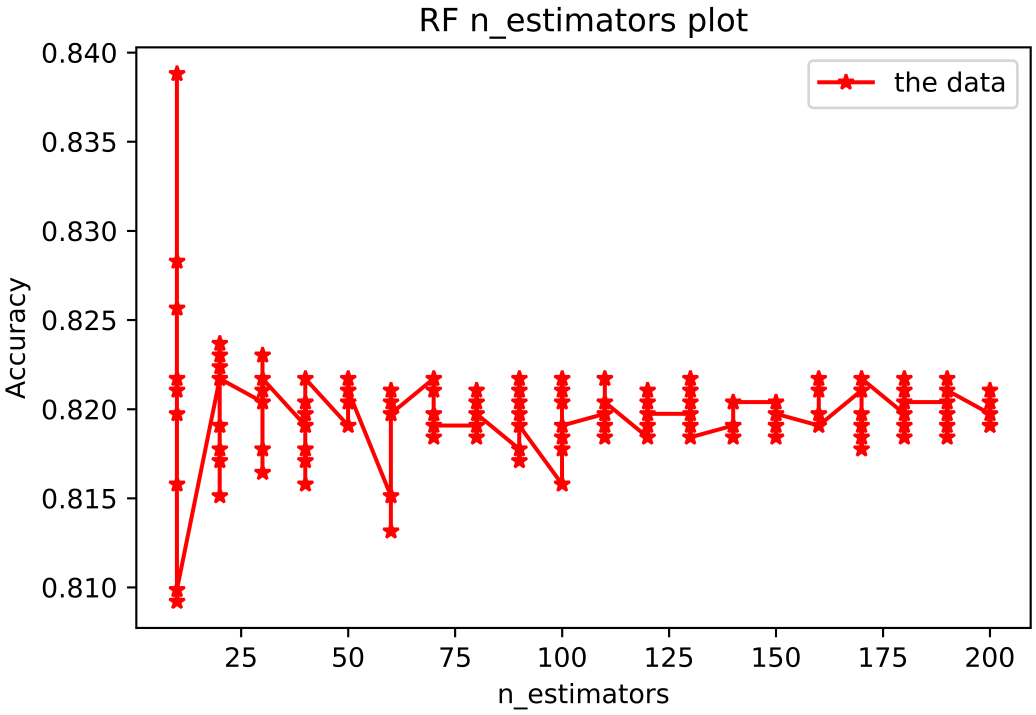

Figure 1: Accuracy vs N_estimators

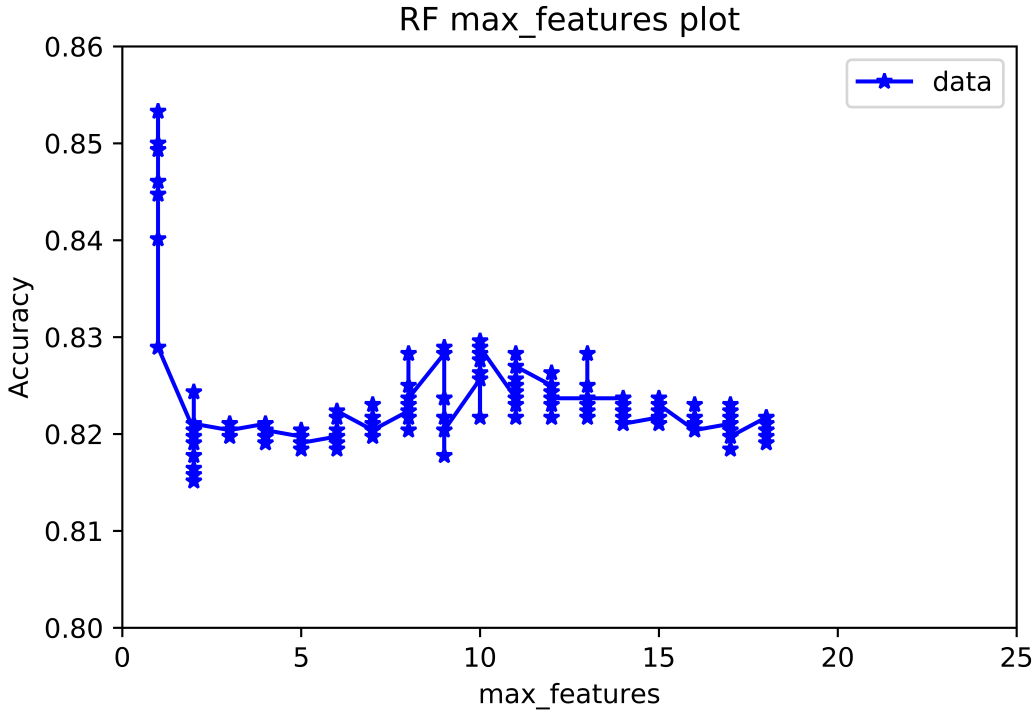

Figure 2: Accuracy vs Max_features

Max_features was also compared against accuracy and from Figure 2 we can see that the max_features should be around 8 or 7. This process of fine tuning is very slow and sometimes imprecise. To speed up the process GridSearchCV is used. GridSearchCV implements an exhaustive search over the hyper parameters to tune. The results of GridSearchCV indicated that max_features should be auto, criterion should be entropy and bootstrap set to true.

K nearest neighbor has only 1 major hyper-parameter to tune and that is, the n_neighbours of indicates the number of neighbours. The n_neighbours were varied from 1-30. The results in Figure 3 show that n_neighbours 3 and 5 both produce best accuracy. n_neighbours was set to 5.

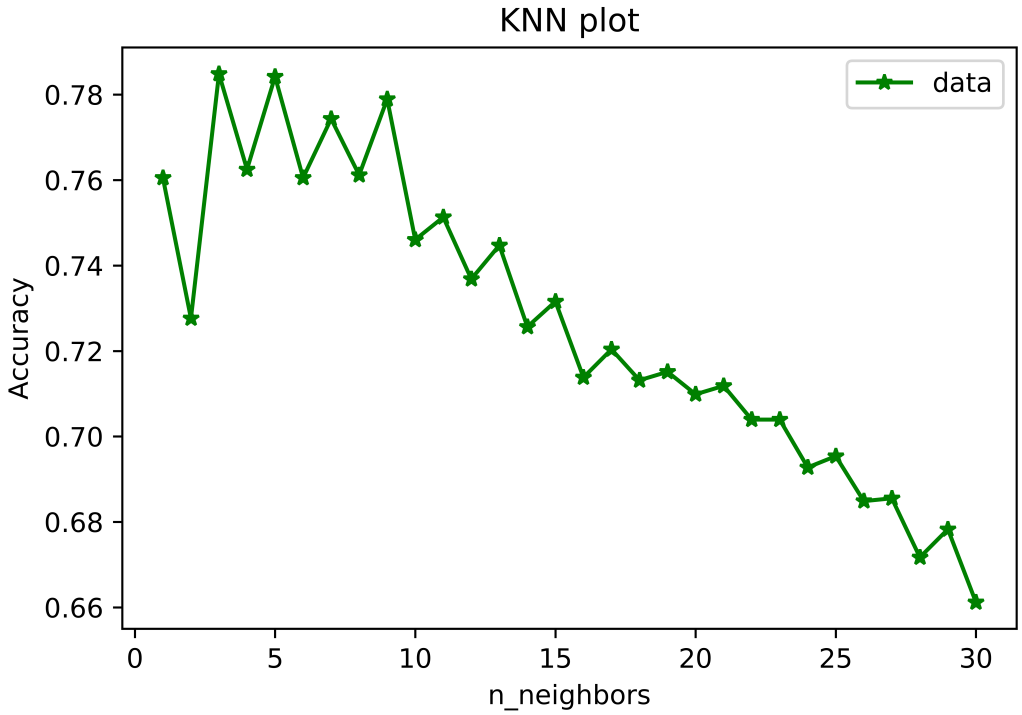

Figure 3: Accuracy vs Max_features

There is associated code for future work in the appendix.

0.3 Cross Validation Protocol

Training and testing relied on a cross-validation protocol where confirm this number 20 proportional testing sets were used such that every data point was used 19 times for training and once for testing. The same data treatment is recommended for data compressed via

Table S1: Classifier Performance Results: Mean and Standard Deviation (σ)

	KNN		RF		Deep Neural Network	
	Mean	σ	Mean	σ	Mean	σ
Accuracy	0.8592	0.0473	0.8801	0.05	82.2943	6.3212

autoencoder.

0.4 ANALYTICAL STRATEGY

After classifiers were evaluated they were subject to the following statistical analysis: The distributions of precision, sensitivity, and f-measure for each attack condition were subject to repeated measures analysis of variance (ANOVA). This was used to determine statistically significant differences in sensitivity, precision, and f-measure using algorithm as the single factor. The significance threshold was determined by applying a Bonferonni correction for multiple comparisons [5] to $\alpha = 0.05$ resulting in a threshold value condition of $p = 0.05/3 = 0.01666$.

It is recommended that in future work the same data preparation, tuning, and cross validation procedure be applied to data compressed using the autoencoder in the code associated with this paper. In that case the analytically strategy recommended would be to apply mixed factorial analysis of variance (ANOVA) using algorithm and dimensional reduction strategy as factors. The number of comparisons should remain the same (3: precision, sensitivity, and f-measure), such that the decision threshold would be $p = 0.05/3 = 0.01666$ as above.

RESULTS

Table 1 shows the mean and standard deviation of the accuracy result of cross validation testing for each algorithm used. ANOVA revealed results to be significant with a p-value<.00001. indicating that the random forest method was the most effective general classification method of those assessed. It should be noted that as these scores are the result of testing on unseen test data, it is possible to train to higher accuracy levels (above 90%) but those are not actually reflective of true classifier performance in the world.

CONCLUSIONS

The finding that random forest outperformed a deep neural network with significance supports the finding of [20] that random forest is the most effective classifier on a variety of IDS problems. Because part of the advantage of deep learning methods is that they can be highly tuned to specific problems we expect they may be more valuable in binary classification targeted to particular attack vectors. For example highly tuned recurrent neural networks (RNNs) may be effective in DDoS classification. It may also be possible to employ hybrid models, much like autoencoders, where deep neural networks are trained to identify some feature not immediately present in data for use by a random forest classifier. Because we designed our classifiers for general rather than binary classification, the individual classification measures are worse than those found by binary classification. This is to be expected as increasing the number of possible labels and corresponding number of features forces the classifier to engage with more noise to learn the useful data about any given class. Nonetheless this approach allows for classification of a unique attack type, rather than conflicting positives. The benefit is that for anyone operating an IDS the clear single label will allow more precise targeting of countermeasures.

It was found that once the classifiers were fully trained they would not make predictions of the DDOS attack vector. It is hypothesized that this is the result of randomizing the order of the data points. Because DDOS is an attack that takes place over many attacks from different IP addresses, the signal of the attack is assumed to be weak in any individual event pattern when compared to the other attack vectors considered. As such the classifiers generally optimize to avoid DDoS predictions because of the high degree of relative uncertainty. [13] had success identifying DDoS with a recurrent neural network (RNN) structure that remembers past events, though the RNN had little success identifying other attacks. It is recommended that in practice a dedicated RNN should be applied to identifying DDoS attempts while another classifier such as a fully connected deep neural network is used to identify other attacks.

The following future work is recommended: The code submitted alongside this paper supports the application of an autoencoder to do a mixed factorial study as described above. This would be of great value as it would help to elucidate whether the superior performance of self-taught learning found by [13] was simply a result of having applied feature selection. Further a clear understanding of effective feature selection for deep learning would inform fair comparison between deep learning and other classification methods. Another possibility is to compare different dimensional reduction techniques during pre-processing of data from those attack vectors represented by too few data points to be used with deep learning.

This could help inform the tuning of classifiers designed to identify patterns of less common attacks.

APPENDIX

Please see attached .zip file for all associated computer code written for use with this project. Authorship of computer code is as follows: John Beninger: DeepNeuralNet.py, autoencoder.py, bigMerge.py, normalization.py, getFormats.sh, findReplaceLabels.sh, runMerge.sh Omar Irfan: randomForest.py, mnb.py, knn.py, qda.py, removeNaN.py, rfgridsearch.py Ahsen Hussain: calcRatios.py, checkAllFloat.py, featureSelect.py, shuffle.py, sorter.py, stringMatch.py. Note that editing and debugging responsibilities were often shared even though the above only indicates the primary author(s).

Bibliography

[1] Abdulrahman Alharby and Hideki Imai. Ids false alarm reduction using continuous and discontinuous patterns. In *International Conference on Applied Cryptography and Network Security*, pages 192–205. Springer, 2005.

[2] Shadi Aljawarneh, Monther Aldwairi, and Muneer Bani Yassein. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, 25:152–160, 2018.

[3] Muder Almi'ani, Alia Abu Ghazleh, Amer Al-Rahayfeh, and Abdul Razaque. Intelligent intrusion detection system using clustered self organized map. In *Software Defined Systems (SDS), 2018 Fifth International Conference on*, pages 138–144. IEEE, 2018.

[4] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2012.

[5] Olive Jean Dunn. Multiple comparisons among means. *Journal of the American statistical association*, 56(293):52–64, 1961.

[6] Chuck Easttom. *Computer security fundamentals*. Pearson Education, 3 edition, 2016.

[7] Nabila Farnaaz and MA Jabbar. Random forest modeling for network intrusion detection system. *Procedia Computer Science*, 89:213–217, 2016.

[8] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[9] Mohammad Sazzadul Hoque, Md Mukit, Md Bikas, Abu Naser, et al. An implementation of intrusion detection system using genetic algorithm. *arXiv preprint arXiv:1204.1336*, 2012.

[10] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 21–26. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016.

[11] Fangjun Kuang, Weihong Xu, and Siyang Zhang. A novel hybrid kpca and svm with ga model for intrusion detection. *Applied Soft Computing*, 18:178–184, 2014.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[13] Brian Lee, Sandhya Amaresh, Clifford Green, and Daniel Engels. Comparative study of deep learning models for network intrusion detection. *SMU Data Science Review*, 1(1):8, 2018.

[14] Shweta Malhotra, Vikram Bali, and KK Paliwal. Genetic programming and k-nearest neighbour classifier based intrusion detection model. In *Cloud Computing, Data Science & Engineering-Confluence, 2017 7th International Conference on*, pages 42–46. IEEE, 2017.

[15] Ishfaq Manzoor, Neeraj Kumar, et al. A feature reduced intrusion detection system using ann classifier. *Expert Systems with Applications*, 88:249–257, 2017.

[16] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *Military Communications and Information Systems Conference (MilCIS), 2015*, pages 1–6. IEEE, 2015.

[17] Sasanka Potluri and Christian Diedrich. Accelerated deep neural networks for enhanced intrusion detection system. In *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*, pages 1–8. IEEE, 2016.

[18] MR Gauthama Raman, Nivethitha Somu, Kannan Kirthivasan, Ramiro Liscano, and VS Shankar Sriram. An efficient intrusion detection system based on hypergraph-genetic algorithm for parameter optimization and feature selection in support vector machine. *Knowledge-Based Systems*, 134:1–12, 2017.

[19] R Ravinder Reddy, Y Ramadevi, and KV N Sunitha. Effective discriminant function for intrusion detection using svm. In *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*, pages 1148–1153. IEEE, 2016.

[20] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, pages 108–116, 2018.

[21] Mansour Sheikhan, Zahra Jadidi, and Ali Farrokhi. Intrusion detection using reduced-size rnn based on feature grouping. *Neural Computing and Applications*, 21(6):1185–1190, 2012.

[22] Ningxin Shi, Xiaohong Yuan, and William Nick. Semi-supervised random forest for intrusion detection network, 2017.

[23] Saeid Soheily-Khah, Pierre-François Marteau, and Nicolas Béchet. Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: A case study on the iscx dataset. In *Data Intelligence and Security (ICDIS), 2018 1st International Conference on*, pages 219–226. IEEE, 2018.

[24] Gina C Tjhai, Maria Papadaki, SM Furnell, and Nathan L Clarke. Investigating the problem of ids false alarms: An experimental study using snort. In *IFIP International Information Security Conference*, pages 253–267. Springer, 2008.

[25] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5:21954–21961, 2017.