Evaluation of machine learning algorithms for anomaly detection

Elmrabit, Nebrase; Zhou, Feixiang; Li, Fengyin; Zhou, Huiyu

Published in:
2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)

DOI:
10.1109/CyberSecurity49315.2020.9138871

Publication date:
2020

Document Version
Peer reviewed version

Link to publication in ResearchOnline

Citation for published version (Harvard):
Elmrabit, N, Zhou, F, Li, F & Zhou, H 2020, Evaluation of machine learning algorithms for anomaly detection. in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security).* IEEE. https://doi.org/10.1109/CyberSecurity49315.2020.9138871

Evaluation of Machine Learning Algorithms for Anomaly Detection

Nebrase Elmrabit
Department of Cyber Security
Glasgow Caledonian University
Glasgow, UK
nebrase.elmrabit@gcu.ac.uk

Feixiang Zhou
School of Informatics
University of Leicester
Leicester, UK
fz64@leicester.ac.uk

Fengyin Li
School of Information Science
Qufu Normal University
Rizhao 276826, China
lfyin318@126.com

Huiyu Zhou
School of Informatics
University of Leicester
Leicester, UK
hz143@leicester.ac.uk

***Abstract*—Malicious attack detection is one of the critical cyber-security challenges in the peer-to-peer smart grid platforms due to the fact that attackers' behaviours change continuously over time. In this paper, we evaluate twelve Machine Learning (ML) algorithms in terms of their ability to detect anomalous behaviours over the networking practice. The evaluation is performed on three publicly available datasets: CICIDS-2017, UNSW-NB15 and the Industrial Control System (ICS) cyber-attack datasets. The experimental work is performed through the ALICE high-performance computing facility at the University of Leicester. Based on these experiments, a comprehensive analysis of the ML algorithms is presented. The evaluation results verify that the Random Forest (RF) algorithm achieves the best performance in terms of accuracy, precision, Recall, F1-Score and Receiver Operating Characteristic (ROC) curves on all these datasets. It is worth pointing out that other algorithms perform closely to RF and that the decision regarding which ML algorithm to select depends on the data produced by the application system.**

***Index Terms*—Cyber Security, intrusion detection, anomaly detection, machine learning, deep learning, smart grid.**

I. Introduction

Anomaly detection in the context of cyber-security is a continuous challenge to our human community, where attackers have the motivation, opportunity and capability to launch their attacks, which can include botnet, brute force, Denial-of-Service (DoS), Distributed Denial-of-Service (DDOS), port scan, Man-in-the-Middle (MITM), Structured Query Language (SQL) injection, and privilege escalation attacks.

The first Intrusion Detection System (IDS) for the identification of anomalous behaviours in network systems was proposed by Anderson [1] in 1980. Anomaly detection methods such as signature-based approaches are usually used to monitor network activities using pre-identified cyber-security attack indicators to specify the security threats that may affect systems' Confidentiality, Integrity or Availability (CIA) [2]. These systems, however, have fatal limitations when it comes to zero-day attacks or encrypted traffic generated by attackers [3]. In the zero-day attacks, where no vulnerability has been previously discovered, no signature will be available to help detect the attacks. On the other hand, it will not be possible for the signature-based approaches to inspect encrypted traffic generated by the attacker if the encryption key is unknown.

It is therefore necessary to consider a defence system to reduce the accompanying risks. This can be achieved by predicting the anomaly behaviours of malicious attacks based on state-of-the-art Machine Learning (ML) algorithms, which are used to train the defence model and predict any anomalous behaviour. It is difficult, however, to achieve a high level of prediction accuracy while maintaining a low false alarm rate, and this challenge remains to be solved in the literature.

Currently, with the use of Artificial Intelligence (AI) technologies, researchers can classify data collected from network activities, Internet of Things (IoT) devices, SCADA systems [4], mobile phones [5], smart grids [6] or log data from any machine into either a binary classification of normal or abnormal behaviours, or multi-class attack classifications of normal or different types of abnormal behaviours associated with specific attacks. Anomaly detection, also called outlier detection, is the method that is used to distinguish the rare events which look unusual from the majority of the data.

The main contributions of this paper are summarised as follows: (1) We review the currently available datasets that have the most the up-to-date attack scenarios. (2) We review the current works that apply ML to the detection of anomalous behaviours. (3) We apply twelve ML algorithms to the selected datasets and evaluate them using binary classification and multi-classification based on the performance metrics. (4) We recommend the best-fit algorithms for the anomaly detection challenge. This paper provides the research community and the cybersecurity industry with insightful knowledge and suggestions regarding suitable ML algorithms to the cybersecurity challenge.

The rest of this paper is structured as follows. In Section II, we review the relevant work in the area of anomaly detection using ML. In Section III, we describe the methodology used in our work and the structure of these methods. Section IV provides a detailed experimental design that includes the running environment, the dataset selected to evaluate the methods and the experimental results for the binary and multi-class attacks classifications. Finally, Section V concludes this paper.

This work was funded by EU Horizon 2020 DOMINOES Project (Grant Number: 771066).

II. BACKGROUND AND RELATED WORK

We first review previous works focusing on ML algorithms and their application to the detection of anomalous behaviours in the cybersecurity field.

A. Machine Learning

In recent years, ML algorithms have become a popular problem-solving approach in various disciplines of science, from computer, vision and behaviour analysis [7] to cyber-security, e.g. anomaly detection [8]. Furthermore, the potential use of ML in different applications looks promising [9], although deep learning also offers promising solutions from a different angle. Since deep learning algorithms require a large dataset for training, however, Parampottupadam and Moldovann [10] conclude that the performance of deep learning models may not necessarily outperform other traditional ML models in some applications.

ML algorithms use statistical models that provide systems with the ability to produce predictions without human intervention, having accessed sample data, known as training data, in order to learn and improve human experiences. Available ML algorithms enable the analysis of large volumes of data, including: supervised ML algorithms, unsupervised ML algorithms, semi-supervised ML algorithms and reinforcement ML algorithms.

In this study, we evaluate six classical supervised ML algorithms and six deep learning algorithms. The selection of these methods is based on the use and the performance of the algorithms in previous studies. The selected classical ML algorithms are Logistic Regression (LR), Gaussian Naive Bayes (GNB), K-nearest Neighbours (KNN), Decision Tree (DT), Adaptive boosting (AdaB), and Random Forest (RF). Our selected deep learning algorithms include Convolutional Neural Network (CNN), Convolutional Neural Network and Long short-Term Memory (CNN-LSTM), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Simple Recurrent Neural Network (RNN), and Deep Neural Network (DNN).

B. Applications of ML Algorithms

There has been an increase in the amount of research in recent years seeking to apply ML to the detection of anomalous behaviours in the cyber-security field, either classical ML classification and deep learning classification techniques, or a combination of both the groups. Vinayakumar et al. [11] proposed a highly scalable and hybrid DNNs framework using KDDCup-99, NSL-KDD, UNSW-NB15, WSN-DS and CICIDS2017 datasets. With five DNN hidden layers, they achieved binary classification accuracy of 92.7%, 78.9%, 76.1%, 98.2% and 93.1% respectively.

Zhang et al. [12] proposed a network intrusion detection based on a deep hierarchical network and original flow data, using CNN classification to learn spatial features and Long short-term memory (LSTM) classification to learn temporal features, for the CICIDS2017 and CTU datasets. After the classification process by CNN+LSTM, 99.8% accuracy was achieved for the CICIDS2017 dataset, and 98.7% accuracy for the CTU dataset.

Al-Zewairi et al. [13] evaluated ANN on the UNSW-NB15 dataset using the back propagation and stochastic gradient descent methods, and delivering the evaluation accuracy of 98.99% with a low false alarm rate of around 0.5%.

Beluch et al. [14] conducted performance evaluation of intrusion detection based ML classification algorithms, i.e. Decision Tree, Support Vector Machine (SVM), RF and Naive Bayes, using big data technologies (Apache Spark [1]) on the full UNSW-NB15 dataset, achieving 97.49 % accuracy.

Faker et al. [15] evaluated the performance of three classification methods (DNN, RF, and GBT) using Apache Spark on subsets of the selected features from both UNSW-NB15 and CICIDS2017 datasets. Both binary and multi-class attack classifications were used in their study, and they concluded that DNN has the best performance, exhibiting 99.19% accuracy using the UNSW-NB15 dataset and 99.99% accuracy using CICIDS2017.

This brief review of the literature shows that there are numerous ways to integrate the use of AI with cyber-security. Using the publicly available datasets, researchers can evaluate and compare the performance of various ML algorithms. This will lead to a better understanding of the topic and provide an excellent chance to implement these technologies within industrial systems to reduce the risk of known and unknown attacks.

III. METHODOLOGY

In this paper, we evaluate twelve ML algorithms in terms of their ability to detect anomalous behaviours that take place in a host or network system to discover possible attacks including DoS, port scanning, SQL injection, brute force, worms and other associated vulnerabilities found in UNSW-NB15 and CICIDS2017 datasets. Moreover, using the Industrial Control Systems (ICS) cyber-attack dataset, we wish to detect anomalous behaviours including data injection, remote tripping command injection, and relay setting change.

In this section, we briefly describe the twelve selected ML algorithms. The selection was based on the popularity and the performance of these algorithms in previous studies.

There is a wide range of ML classification techniques, which, although they share the same objectives, encompass very different mathematical models, strengths and weaknesses. Each technique attempts to classify data in different circumstances. K-nearest neighbours (KNN) [16], for example, is a traditional non-parametric technique to classify samples. KNN is a type of instance-based learning algorithm. This classification method is based on a distance function that measures the estimated distances between objects to assign all unlabelled objects to the most common among its K nearest neighbours, where the K value is always a positive integer number. Naive Bayes [17] is a traditional classifier that applies Bayes' theorem of pre-probability to classify data instances to

[1] https://spark.apache.org/

TABLE I: The advantages and disadvantages of the selected machine learning algorithms.

Algorithm	Advantages	Disadvantages
Logistic Regression (LR)	Low computing cost and fast speed. Its output can be interpreted as a probability	May suffer from under-fitting. The performance is poor when the feature space is large
Gaussian Naive Bayes (GNB)	Fast training speed for small and big datasets. Less sensitive to missing data	It needs to calculate prior probabilities. It does not work well if the sample's attributes are related
K-nearest Neighbours (KNN)	It can be used for both classification and regression. Easy to understand and implement	Poor performance when analysing unbalanced samples. High computational complexity for large datasets
Decision Tree (DT)	Fast prediction. Addressing highly non-linear data	Suffere from over-fitting. More time is needed to train the model
Adaptive Boosting (AdaB)	Various algorithms can be used to build sub-classifiers. Does not easily lead to over fitting	The performance depends on the selected weak classifier. Sensitive to outliers
Random Forest (RF)	Robust to outliers and can handle them well. It is comparatively less affected by noise	Longer training time since it will generate many trees. Much more computational power and resources needed
Convolutional Neural Network (CNN)	Handle high-dimensional data well with shared convolution kernels	The training result easily converges to the local minimum instead of the global minimum using the gradient descent
Long Short-Term Memory (LSTM)	Gate mechanism greatly alleviates the problem of gradient disappearance and simplifies the complexity of tuning parameters	High computing cost
Gated Recurrent Units (GRU)	Gate mechanism alleviates the problem of gradient disappearance	High computational cost
Simple Recurrent Neural Network (RNN)	It can learn and use contextual information explicitly in sequence prediction	The problem of gradient disappearance occurs readily
Deep Neural Network (DNN)	It can execute feature engineering on it own compared with traditional ML methods	Many parameters need to be fine-tuned

a specific class. Logistic Regression [18] is another classifier, based on a logistic regression function also called a sigmoid function. The decision trees method [19] classifies data into branch segments including a root node, internal nodes and leaf nodes. Random Forests (RF) [20] classification works by joining several decision trees together to correct over-fitting to the training set of the decision trees. AdaBoost [21], another classic ML classification approach, works by creating a strong classifier from a number of weak classifiers.

On the other hand, in recent years, a number of deep learning classification techniques have been developed. These use multiple hidden layers between the input and output progressively to extract higher-level features from large datasets. Deep Neural Network (DNN) can deal with a complex non-linear relationship, and is a feed-forward network where the direction of data is from the input layer to the output layer. An opposite approach is followed by the Simple Recurrent Neural Network (RNN) [22] and Long Short-Term Memory (LSTM) [23] where data can flow in both directions. Convolutional Neural Network (CNN) [24] is a deep learning architecture where each neuron is connected to all the other neurons at the following layer by using smaller and simpler patterns. This technique allows one to build more complex patterns. Finally, Gated Recurrent Units (GRU) [25] are similar to LSTM with a forget gate added to the memory block in the LSTM architectures. GRU has fewer parameters than LSTM, which leads to better performance in various tasks. Table 1 summarises the advantages and disadvantages of the selected ML algorithm.

Two types of label classification methods are used in our study, i.e. binary attack classification and multi-class attack classification. The general structure of the proposed method shown in Fig. 1 is divided into four phases as follows: The dataset processing phases, where we select the right dataset for the model. The dataset reprocessing phase, where data is integrated and filtered before feature scaling or normalisation is applied, moving then to binary or multi classification labelling. The next phase is ML algorithm analysis, using one of the selected algorithms to train and test the data. In the final stage, we apply a number of performance metrics to measuring and evaluating the performance of the used algorithm. More details are presented and discussed in the experiment section IV.

IV. EXPERIMENTS

This section presents the experimental design in detail, which includes the environment used to run the discussed methods and the general structure of the experimental design. Also, we introduce the dataset used to evaluate the methods and pre-processing phases. Finally, the experimental results are presented, which includes binary and multi-class classification.

A. Experimental Design

The experiments were performed through the ALICE high-performance computing facility at the University of Leicester[2]. Each computer had 64GB RAM, two Ivy Bridge CPUs at 2.50GHz (20 cores in total) and 2 x Nvidia Tesla P100 GPU cards. Using Python-3.6.8 running on the Community

[2] https://www2.le.ac.uk/offices/itservices/ithelp/services/hpc

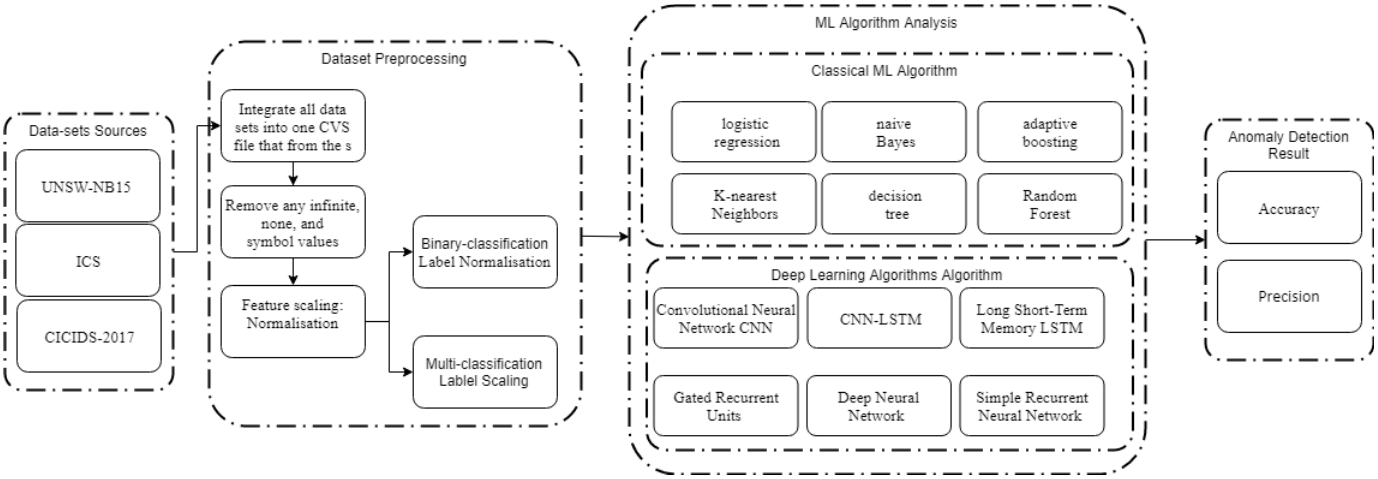

Data-sets Sources
UNSW-NB15
ICS
CICIDS-2017
Dataset Preprocessing
Integrate all data sets into one CVS file that from the s
Remove any infinite, none, and symbol values
Feature scaling: Normalisation
Binary-classification Label Normalisation
Multi-classification Lablel Scaling
ML Algorithm Analysis
Classical ML Algorithm
logistic regression
naive Bayes
adaptive boosting
K-nearest Neighbors
decision tree
Random Forest
Deep Learning Algorithms Algorithm
Convolutional Neural Network CNN
CNN-LSTM
Long Short-Term Memory LSTM
Gated Recurrent Units
Deep Neural Network
Simple Recurrent Neural Network
Anomaly Detection Result
Accuracy
Precision

Fig. 1: General structure of the proposed method.

Enterprise Operating System[3] (CentOS Linux 7). The classical ML algorithms were implemented using Scikit-learn-0.21.3 ML library. The Deep learning algorithms were implemented using Keras-2.3.0[4] neural-network library on top of TensorFlow-1.9.0[5] to enable the use of GPU. Sigmoid and softmax functions were also used for binary and multi-class classification. Furthermore, Pandas[6] and NumPy[7] library packages were used to manipulate and analyse the raw data.

B. Datasets and Data Pre-processing

Numerous numbers of network-based datasets are available online for the research community to train and evaluate their approaches to the detection of anomalous network behaviours on their platforms. Finding the right dataset that contains up-to-date attack scenarios such as botnet, brute force, Dos, DDOS, port scan, CCS, SQL injection and privilege escalation, with the right dataset properties in terms of usability format and labelling results, is not an easy task.

We started by looking into the most widespread dataset for IDS, called the KDD CUP 99 [26], which includes over five million data points with twenty attack scenarios. Later on, Tavallaee et al. [27] enhanced KDD CUP 99 by removing redundancy records, namely NSL-KDD 2009. We believe this dataset is outdated as the network traffics were generated in 1998, which cannot reflect the new network structures and dynamics of attacks.

Ring et al. [28] comprehensively reviewed 34 datasets by pointing out the characteristics of each dataset. Based on their discussion and our assessment, CICIDS2017 [29] and UNSW-NB15 [30] were used to evaluate the twelve ML algorithms in our current paper. This is because these datasets contain a wide range of current attack scenarios, which meet the real-world criteria. They are also publicly available. On the other hand, the power system ICS cyber-attack dataset [31] was also used since it reflects the categories of the attacks in power system platforms. However, opposite to the previous two datasets, it is noticed that the percentage of the attack instances are higher than that of normal behaviour instances. Because the power system dataset has just eight natural event scenarios compared to twenty-eight attack events scenarios. Finally, the three selected datasets have different properties such as event types, attack labels names, features collected, duration and scenarios, which are summarised in Table II.

TABLE II: Datasts Propertties

Records	**Dataset**		
	UNSW-NB15	**CICIDS-2017**	**ICS cyber-attack**
Total	2,540,047	2,830,743	78,391
Attacks	321,283 12%	557,646 19.7%	55,677 71%
Normal	2,218,764 88%	2,273,097 80.3%	22714 29%
Features	49	78	128
Duration	31 Hours	5 Days	37 scenarios

Data cleaning techniques was performed through four main stages:

1) Convert and integrate all the files from the same dataset to one single CSV file.
2) Delete any infinite, none, and symbol values.
3) Feature scaling by normalising all the features.
4) The final step depended on the label classification types that we used; namely binary classification used label normalisation and multi-class classification used label scaling.

C. Performance Metrics

Seven evaluation matrices are used to measure the performance of the selected ML algorithms, i.e. accuracy, precision, True Positive Rate (TPR), also known as Recall, False Positive Rate (FPR), F1-Score, Receiver Operating Characteristic

[3] https://www.centos.org/
[4] https://keras.io/
[5] https://www.tensorflow.org/
[6] https://pandas.pydata.org/
[7] https://numpy.org/

(ROC) curve and Confusion Matrix. These are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

$$F1 - score = 2 \times \left(\frac{Precision \ \times Recall}{Precision + Recall} \right)$$

$$AUC = \int_0^1 \frac{TP}{TP + FN} d \, \frac{FP}{TN + FP}$$

Accuracy refer to the percentage of the total number of correct classifications, while Precision refers to the closeness of the measurements to each other. Recall measures the percentage of actual positives which are classified as attacks. FPR measures the percentage of normal traffic flagged as attacks to normal connection data. The F1-score is a measure of the test accuracy. The Area Under the ROC Curve (AUC) summarises the size of the area under the ROC curve, which is the trade-off between the TPR and FPR using a different probability threshold. If the accuracy, precision, recall, F1-score and AUC are higher, the machine learning model is better. On the contrary, if the FPR is lower, the model is better.

On the other hand, True Positive (TP) refers to the percentage of attack traffic correctly classified as attack data. False Negative (FN) refers to the percentage of attack traffic incorrectly classified as normal data. False Positive (FP) refers to the percentage of the normal traffic incorrectly classified as attack data. True Negative (TN) refers to the percentage of the normal traffic correctly classified as normal data.

Confusion Matrix is used to evaluate the performance based on the capability of classifying network traffic into a correct attack type. A confusion matrix allows the visualisation of the agreement between the true label and the prediction label.

D. Experimental Results

Twelve ML algorithms are applied in this phase, six of them using the classical ML algorithms and the rest using deep learning algorithms. These, have been introduced in the previous sections. The three datasets were separated so that 70% of the data is used as a training dataset, used to train the ML models. The remaining 30% of the data form the test dataset, which is used to evaluate the trained ML models [32]. Split the dataset to 70:30 ratio proves a better result than the other tested ratios in our cases. Two classifications were used to evaluate the datasets. First, the binary classification where the label has just two outcomes, normal or attack; second, the multi-class classification labels have a range of values that may be assigned based on attacks types.

TABLE III: Binary Classification

Methods	Accuracy	Precision	Recall	F-score
UNSW-NB15				
LR	0.753	0.858	0.735	0.792
GNB	0.716	0.693	0.997	0.818
KNN	0.829	0.851	0.887	0.869
DT	0.885	0.914	0.906	0.910
AdaB	0.839	0.817	0.965	0.884
RF	0.877	0.844	0.991	0.912
CNN	0.856	0.825	0.983	0.897
CNN-LSTM	0.835	0.804	0.980	0.889
LSTM	0.767	0.893	0.721	0.798
GRU	0.777	0.857	0.782	0.818
SimpleRNN	0.807	0.775	0.984	0.867
DNN	0.827	0.793	0.987	0.879
CICIDS-2017				
LR	0.883	0.737	0.634	0.682
GNB	0.550	0.298	0.946	0.453
KNN	0.996	0.987	0.994	0.990
DT	0.998	0.995	0.996	0.996
AdaB	0.962	0.898	0.910	0.904
RF	0.999	0.997	0.997	0.997
CNN	0.996	0.991	0.989	0.990
CNN-LSTM	0.993	0.989	0.992	0.991
LSTM	0.994	0.967	0.961	0.964
GRU	0.994	0.981	0.989	0.989
SimpleRNN	0.983	0.965	0.951	0.958
DNN	0.991	0.976	0.987	0.981
ICS cyber-attack datasets				
LR	0.710	0.710	1.000	0.830
GNB	0.709	0.710	0.999	0.830
KNN	0.849	0.882	0.909	0.895
DT	0.864	0.905	0.903	0.904
AdaB	0.720	0.732	0.956	0.829
RF	0.928	0.929	0.972	0.950
CNN	0.715	0.715	0.999	0.834
CNN-LSTM	0.715	0.715	1.000	0.833
LSTM	0.715	0.715	1.000	0.833
GRU	0.715	0.715	1.000	0.834
SimpleRNN	0.715	0.715	0.999	0.834
DNN	0.716	0.716	1.000	0.834

1) Binary Classification, the prediction accuracy for UNSW-NB15 dataset reached 88.5% using the Decision tree algorithm, while the other algorithms are in the range of 71% to 87.7%. Meanwhile, the accuracy for the CICIDS-2017 dataset reaches 99.9% using the RF algorithm, whereas the other algorithms are in the range of 55% to 99.8%. On the other hand, when using the ICS cyber-attack datasets, RF achieves the best result, with the accuracy of 92.8%, whereas other algorithms are in the range of 70% to 86%. Both the RF and Decision tree accuracy results with respect to binary classification are better than the other methods for all the datasets. This indicates that the DT and RF classifiers are relatively generalisable and can detect new attacks effectively. In addition, it is obvious that the performance of deep learning algorithms including CNN, CNN-LSTM, LSTM, GRU, SimpleRNN and DNN in ICS cyber-attack datasets is worse than that of KNN, DT and RF. The main reason is that this dataset is small, and robust deep learning models need to be fed with a great number of data for training.

The detailed results of the binary classification for all the evaluation methods are shown in Table III, encompassing accuracy, precision, recall and f-score.

2) Multi-class Classification, the prediction accuracy for UNSW-NB15 dataset reaches 73.6% using the RF algorithm, while the other algorithms are in the range of 8.5% to 73.5%. Meanwhile, the accuracy for the CICIDS-2017 dataset reaches 99.9% using the RF algorithm, whereas the other algorithms are in the range of 43% to 99.8%. On the other hand, when using the ICS cyber-attack datasets, the DT algorithm reaches 92.4% accuracy, whereas the other algorithms are in the range of 6% to 92%. Similarly, both the RF and DT accuracy results in respect to multi-class classification are better than that of the other methods for all the datasets. Especially in the ICS cyber-attack dataset, the performance of the two classifiers is clearly superior to that of deep learning and other classical machine learning algorithms. The detailed result of the multi-class classification for all the evaluation methods are shown in Table IV, encompassing accuracy, precision, recall and f-score .

E. Discussion

To discuss the results further, we create ROC curves for the UNSW NB-15, CICIDS-2017 and ICS cyber-attack datasets. These are shown in Figs. 2 and 3. Sometimes the differences in the results are marginal, but using the AUC metric, we observe that RF has the best performance result of all the algorithms, ranging from 0.96 to 1 for the binary classification, and from 0.97 to 1 for multi-classification. This indicates that RF achieves the lowest FPR and the highest TPR.

The primary concept behind RF is that it combines a single network of the companion of multi decision trees into a single model. With a large dataset, however, the training time will be very long since it generates many trees that require high computational power and resources. We see that RF outperforms all the evaluated deep learning algorithms in our case, because deep learning algorithms require a larger dataset than that which we use to generate better performance.

On the other hand, Gaussian Naive Bayes has the worst performance of all the tested algorithms, ranging from 0.51 to 0.84 for the binary classification, from 0.54 to 0.86 for multi-classification. In comparison to all the other algorithms, therefore, Naive Bayes achieves the lowest TPR and the highest FPR.

The fact that Gaussian Naive Baies exhibits the worst evaluation performance is not surprising, since it depends heavily on the independence between the input variables, and performance is known to be affected if the sample's attributes are related.

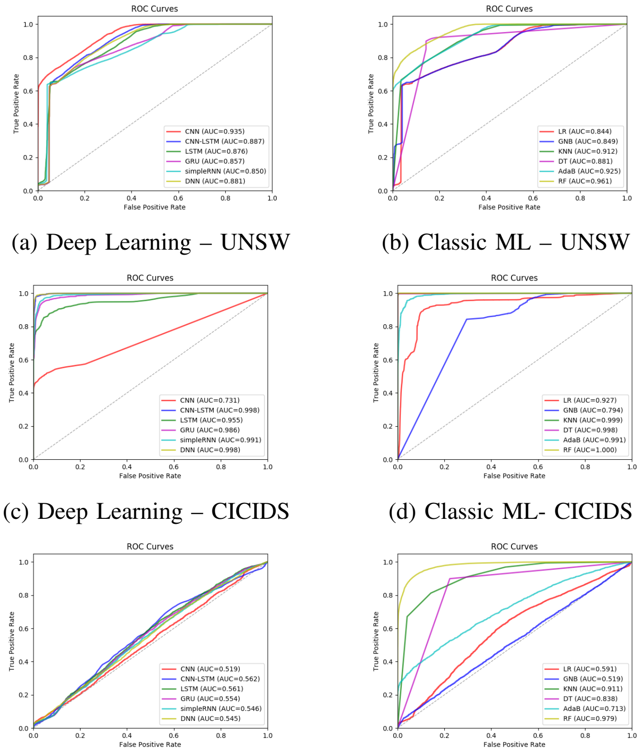
Also, we notice from Figs. 2 (a,c) and 3 (a,e) that integrating CNN, which connects a neuron to all the other neurons at the next layer, with LSTM, where data flows in both directions, does not always serve to improve the performance. We may make the model more sophisticated and in particular, enhance the network bone itself.

TABLE IV: Multi-class Classification

Methods	**Accuracy**	**Precision**	**Recall**	**F-score**
		UNSW-NB15		
LR	0.561	0.497	0.561	0.428
GNB	0.085	0.587	0.085	0.130
KNN	0.652	0.638	0.652	0.638
DT	0.735	0.715	0.735	0.718
AdaB	0.631	0.553	0.631	0.557
RF	0.736	0.726	0.736	0.695
CNN	0.684	0.672	0.684	0.627
CNN-LSTM	0.680	0.619	0.680	0.615
LSTM	0.661	0.601	0.661	0.598
GRU	0.665	0.600	0.661	0.608
SimpleRNN	0.662	0.585	0.662	0.587
DNN	0.663	0.664	0.663	0.608
		CICIDS-2017		
LR	0.915	0.914	0.915	0.910
GNB	0.430	0.846	0.430	0.522
KNN	0.996	0.996	0.996	0.996
DT	0.998	0.998	0.998	0.998
AdaB	0.818	0.769	0.818	0.760
RF	0.999	0.999	0.999	0.999
CNN	0.997	0.996	0.997	0.996
CNN-LSTM	0.994	0.993	0.994	0.994
LSTM	0.991	0.990	0.991	0.989
GRU	0.993	0.993	0.993	0.991
SimpleRNN	0.994	0.993	0.994	0.993
DNN	0.998	0.998	0.998	0.998
		ICS cyber-attack datasets		
LR	0.068	0.036	0.068	0.017
GNB	0.107	0.164	0.107	0.062
KNN	0.877	0.878	0.877	0.877
DT	0.924	0.924	0.924	0.924
AdaB	0.185	0.070	0.185	0.090
RF	0.920	0.920	0.920	0.920
CNN	0.061	0.004	0.061	0.007
CNN-LSTM	0.061	0.004	0.062	0.007
LSTM	0.369	0.307	0.369	0.319
GRU	0.321	0.240	0.321	0.262
SimpleRNN	0.244	0.189	0.244	0.198
DNN	0.379	0.332	0.379	0.308

A multi-classification confusion matrix is used for further investigation into the RF results as shown in Figure 4, where the result shows that RF archives satisfactory prediction results close to the true label for the CICIDS-2017 dataset. The accuracy of detecting Infiltration is much lower than that of other categories because the number of this type of attack is relatively small. Thus it is difficult for the generated model to learn robust feature representation of this attack. However, using UNSW-NB15 dataset, the matrix as shown in Figure 5 shows a mixed result, where, RF has some very low agreement to detect attacks e.g. Back-doors, DoS, and exploitaion.

Finally, the overall performance of ML algorithms using the general-purpose network datasets (UNSW NB-15 and CICIDS-2017) seems better than that of the ICS cyber-attack dataset. This is because the structure of the general-purpose network's datasets are less complicated than the ICS cyber-attack dataset, also, the ICS cyber-attack dataset has 37 scenarios compared with two to nine attack scenarios on the other two datasets.



(a) Deep Learning – UNSW

(b) Classic ML – UNSW

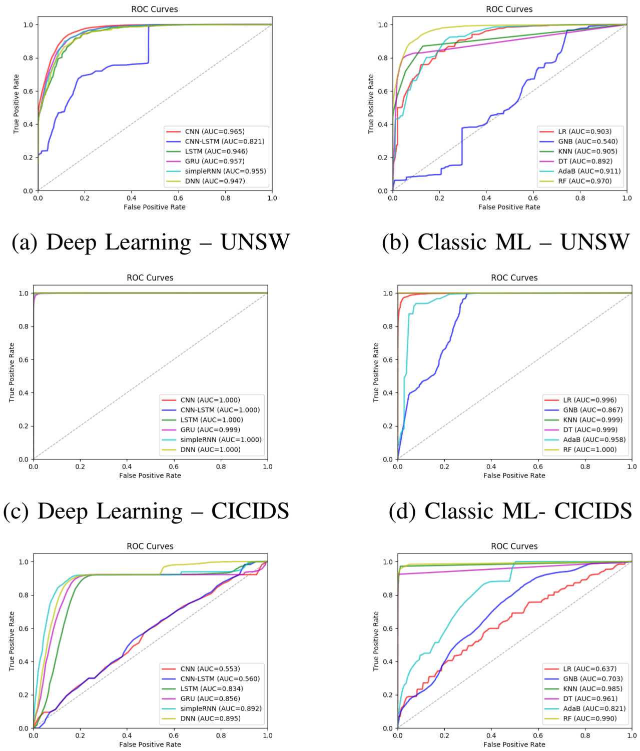
(c) Deep Learning – CICIDS

(d) Classic ML- CICIDS

(e) Deep Learning – ICS

(f) Classic ML- ICS

Fig. 2: Binary Classification ROC Curves



(a) Deep Learning – UNSW

(b) Classic ML – UNSW

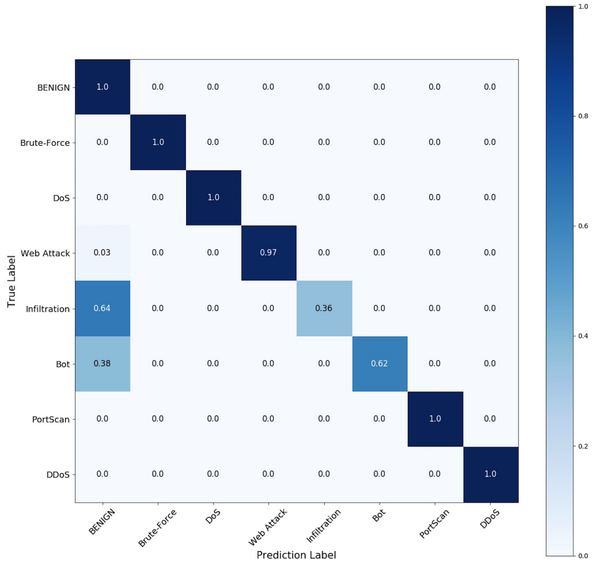
(c) Deep Learning – CICIDS

(d) Classic ML- CICIDS

(e) Deep Learning – ICS

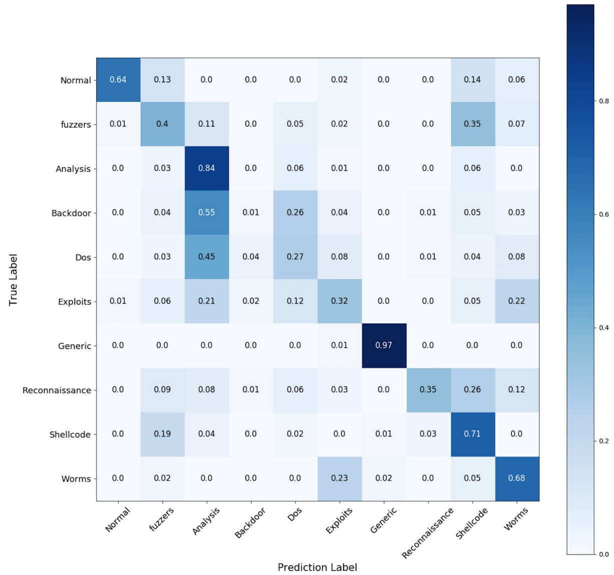
(f) Classic ML- ICS

Fig. 3: Multi Classification ROC Curves



	BENIGN	Brute-Force	DoS	Web Attack	Infiltration	Bot	PortScan	DDoS
BENIGN	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Brute-Force	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
DoS	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Web Attack	0.03	0.0	0.0	0.97	0.0	0.0	0.0	0.0
Infiltration	0.64	0.0	0.0	0.0	0.36	0.0	0.0	0.0
Bot	0.38	0.0	0.0	0.0	0.0	0.62	0.0	0.0
PortScan	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
DDoS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Fig. 4: RF Confusion Matrix Result for CICIDS-2017 Dataset.



	Normal	fuzzers	Analysis	Backdoor	Dos	Exploits	Generic	Reconnaissance	Shellcode	Worms
Normal	0.64	0.13	0.0	0.0	0.0	0.02	0.0	0.0	0.14	0.06
fuzzers	0.01	0.4	0.11	0.0	0.05	0.02	0.0	0.0	0.35	0.07
Analysis	0.0	0.03	0.84	0.0	0.06	0.01	0.0	0.0	0.06	0.0
Backdoor	0.0	0.04	0.55	0.01	0.26	0.04	0.0	0.01	0.05	0.03
Dos	0.0	0.03	0.45	0.04	0.27	0.08	0.0	0.01	0.04	0.08
Exploits	0.01	0.06	0.21	0.02	0.12	0.32	0.0	0.0	0.05	0.22
Generic	0.0	0.0	0.0	0.0	0.0	0.01	0.97	0.0	0.0	0.0
Reconnaissance	0.0	0.09	0.08	0.01	0.06	0.03	0.0	0.35	0.26	0.12
Shellcode	0.0	0.19	0.04	0.0	0.02	0.0	0.01	0.03	0.71	0.0
Worms	0.0	0.02	0.0	0.0	0.0	0.23	0.02	0.0	0.05	0.68

Fig. 5: RF Confusion Matrix Result for UNSW-NB15 Dataset.

V. CONCLUSIONS

In this paper, we have comprehensively evaluated the performance of the twelve ML algorithms for the detection of anomalous behaviours that may be indicative of cyber attacks. In order to recommend the best-fit algorithms, three datasets (i.e. UNSW-NB15, CICIDS-2017, and ICS cyber-attack) were applied to the selected methods. Results in most of the cases show that the RF classification provides the best accuracy, precision, recall and AUC among the methods used in this paper. This may be due to the fact that deep learning classification requires a very large amount of data to train the models and this is not available in the current studies. Meanwhile, as discussed in the previous section, Naive Bayes classification has the lowest performance in terms of accuracy, precision, recall and AUC.

It is evident, however, that the selection of the most appropriate method will depend on the datasets used. We also observe that, in many cases, the differences in the performance table are marginal.

Our future work is to evaluate these methods on the smart grid dataset that we are creating using the same environment. Also, the training and testing time of the selected method will be measured to identify the best performance in terms of efficiency.

References

[1] J. P. Anderson, "Computer security threat monitoring and surveillance," *Technical Report James P Anderson Co Fort Washington Pa*, p. 56, 1980. [Online]. Available: http://www.citeulike.org/user/animeshp/article/592588

[2] M. Swanson, "Security Self-Assessment Guide for Information Technology Systems," *National Institute for Standards and Technology Special Publication*, vol. 800, no. 26, 2001.

[3] N. Moustafa, K. K. R. Choo, I. Radwan, and S. Camtepe, "Outlier Dirichlet Mixture Mechanism: Adversarial Statistical Learning for Anomaly Detection in the Fog," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 1975–1987, 2019.

[4] A. Robles-Durazno, N. Moradpoor, J. McWhinnie, and G. Russell, "A supervised energy monitoring-based machine learning approach for anomaly detection in a clean water supply system," *2018 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2018*, pp. 1–8, 2018.

[5] S. Y. Yerima and S. Khan, "Longitudinal performance analysis of machine learning based Android malware detectors," *2019 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2019*, pp. 1–8, 2019.

[6] M. A. Ferrag and L. Maglaras, "DeepCoin: A Novel Deep Learning and Blockchain-Based Energy Exchange Framework for Smart Grids," *IEEE Transactions on Engineering Management*, vol. PP, pp. 1–13, 2019.

[7] Z. Jiang, D. Crookes, B. D. Green, Y. Zhao, H. Ma, L. Li, S. Zhang, D. Tao, and H. Zhou, "Context-Aware Mouse Behavior Recognition Using Hidden Markov Models," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1133–1148, 2019.

[8] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics (Switzerland)*, vol. 8, no. 3, 2019.

[9] I. Portugal, P. Alencar, and D. Cowan, "The use of machine learning algorithms in recommender systems: A systematic review," *Expert Systems with Applications*, vol. 97, pp. 205–227, 2018.

[10] S. Parampottupadam and A. N. Moldovann, "Cloud-based Real-time Network Intrusion Detection Using Deep Learning," *2018 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2018*, pp. 1–8, 2018.

[11] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.

[12] X. Zhang, J. Chen, Y. Zhou, L. Han, and J. Lin, "A Multiple-Layer Representation Learning Model for Network-Based Attack Detection," *IEEE Access*, vol. 7, pp. 91 992–92 008, 2019.

[13] M. Al-Zewairi, S. Almajali, and A. Awajan, "Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system," *Proceedings - 2017 International Conference on New Trends in Computing Sciences, ICTCS 2017*, vol. 2018-Janua, pp. 167–172, 2017.

[14] M. Belouch, S. El Hadaj, and M. Idlianmiad, "Performance evaluation of intrusion detection based on machine learning using apache spark," *Procedia Computer Science*, vol. 127, pp. 1–6, 2018.

[15] O. Faker and E. Dogdu, "Intrusion Detection Using Big Data and Deep Learning Techniques," in *Proceedings of the 2019 ACM Southeast Conference*. New York, USA: ACM Press, 2019, pp. 86–93.

[16] D. W. Aha, D. Kibler, and A. Marc, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[17] G. H. John and P. Langley, "Estiamting Continuous Distributions in Bayesan Classifiers," *Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, 1995.

[18] S. L. C. Houwelingen and J. C. Van, "Ridge Estimators in Logistic Regression," *Royal Statistical Society*, vol. 43, no. 1, pp. 95–108, 1992.

[19] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[20] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[21] Y. Freund, R. E. Schapire, and M. Hill, "Experiments with a New Boosting Algorithm," *ICML*, 1996.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, oct 1986.

[23] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] S. Lawrence, L. G. C., and C. T. Ah, "Face Recognition: A Convolutional Neural-Network Approach Steve," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, sep 1997.

[25] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1724–1734, 2014.

[26] S. Stolfo, "KDD Cup 1999 Data." [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[27] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, no. Cisda, pp. 1–6, 2009.

[28] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers and Security*, vol. 86, pp. 147–167, 2019.

[29] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*, vol. 2018-Janua, no. Cic, pp. 108–116, 2018.

[30] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*, pp. 1–6, 2015.

[31] U. Adhikari, S. Pan, and T. Morris, "Power System Datasets in Industrial Control System (ICS) Cyber Attack Datasets." [Online]. Available: https://sites.google.com/a/uah.edu/tommy-morris-ua

[32] H. Liu and M. Cocea, "Semi-random partitioning of data into training and test sets in granular computing context," *Granular Computing*, vol. 2, no. 4, pp. 357–386, 2017.