# Enhancing Cyber Defense: Using Machine Learning Algorithms for Detection of Network Anomalies

Zhida Li[1] and Ljiljana Trajković[2]

***Abstract*— Developing advanced cyber defense techniques is essential for effectively detecting network anomalies that are becoming more challenging to identify. In this paper, we generate machine learning models based on real-time Internet and historical data and evaluate their classification performance. We introduce a network anomaly detection tool *CyberDefense* that integrates various stages of the anomaly detection process. It facilitates performance evaluation of machine learning algorithms and generation of new machine learning models. Its modular and scalable design enables incorporating new datasets and machine learning algorithms. The tool has been utilized to generate models and evaluate their classification performance using datasets collected during reported power outage and ransomware attacks.**

## I. Introduction

Network anomalies and their effect on performance of communication networks have dire economic consequences. Identifying these anomalous events and their causes is an important step in preventing anomalous routing that affects performance of the Internet border gateway protocol (BGP). Classification of anomalous events helps alleviate their effects on network performance.

Various network intrusion detection techniques [1]–[3] have been used to detect anomalies such as worms as well as intrusion and distributed denial of service (DDoS) attacks. Network anomalies may be detected by analyzing collected traffic data and generating classification models. Intrusion detection techniques are categorized as anomaly-based, signature-based, multi-agent-based, and hybrid [2]. Anomaly-based detection employs machine learning to identify unexpected events based on statistical patterns inferred from regular behavior. While anomaly-based detection identifies unknown attacks, signature-based detection compares the incoming traffic with the rules and patterns in the databases of known anomalous events. Multi-agent-based detection involves cooperation of multiple agents to detect persistent threats and attacks. Hybrid detection combines multiple techniques to enhance the model robustness and improve the detection accuracy.

Various machine learning algorithms and tools have been used to analyze and classify network anomalies including Internet worms, denial of service attacks, power outages, and ransomware attacks. They are analyzed based on routing and network connection records and transport control and user datagram protocol flows. Machine learning algorithms [4], [5] have been successfully implemented in various intrusion detection systems (IDSs). They include support vector machine, naïve Bayes, decision tree, hidden Markov model, extreme learning machine, multilayer perceptron, convolutional neural networks (CNNs), recurrent neural networks (RNNs), autoencoders, broad learning systems (BLSs), and gradient boosting machines [6]–[9].

Proposed IDSs have been implemented as either real-time or off-line software tools. Snort [10] is an early proposed open-source IDS designed for real-time traffic analysis and monitoring of Internet protocol (IP) networks. It enables analyzing network packets based on predefined rules. Passban [11] is an anomaly-based IDS used to detect malicious attacks (port scanning, HTTP login brute force, SSH login brute force, SYN flood) involving the Internet of things (IoT) devices. The tool relies on isolation forest and local outlier factor lightweight algorithms. VMGuard [12] utilizes random forests to classify hidden malicious processes from tenant virtual machines. SwiftIDS [13] is a near real-time system that employs LightGBM for generating detection models. Real-time data acquisition, data processing, and decision-making phases were implemented using a parallel intrusion detection mechanism. WisdomSDN [14] was proposed to detect the domain name system (DNS) amplification DDoS attack in a simulated software defined network. Commercial tools include BGProtect [15] and various intrusion prevention systems (IPSs) such as Cisco IPS [16], FortiGuard IPS [17], and Palo Alto Networks advanced threat prevention [18]. IP BGP hijacks may be detected and analyzed by using algorithms and tools developed by BGProtect [15]. IPSs enable inspection of traffic flows, prevention of vulnerability exploits, and traffic blocking. They are commonly deployed behind firewalls.

Most IDS tools include models based on specific types of network anomalies and algorithms [11]–[14] while commercial tools are not publicly available [15]–[18]. This void was a motivation to develop the *CyberDefense* [19] anomaly detection tool that facilitates performance evaluation of machine learning algorithms and generation of new machine learning models based on historical and real-time data. *CyberDefense* includes various machine learning algorithms such as long short-term memory (LSTM), gated recurrent unit (GRU), gradient boosting decision trees (GBDT), and newly proposed variable features BLS algorithms without (VFBLS) and with cascades (VCFBLS) [5], [6]. The tool is publicly available for macOS and Linux systems as a web-based version. The tool is used to detect the BGP anomalies

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant R31-611284.

[1]Zhida Li is with New York Institute of Technology, Vancouver, British Columbia, Canada zli74@nyit.edu

[2]Ljiljana Trajković is with the Simon Fraser University, Vancouver, British Columbia, Canada ljilja@sfu.ca

based on routing information service (RIS) raw data collected from Internet exchange points.

The paper is organized as follows: Intrusion detection techniques and tools are introduced in Section I while *CyberDefense* high-level architecture and its implementation are described in Section II. The experimental procedure and performance evaluation are given in Section III. We conclude with Section IV.

## II. CyberDefense Tool

The *CyberDefense* tool integrates various stages of the anomaly detection process that includes data container, real-time data retrieval, off-line data download, feature extraction, label refinement, data partitioning, data processing, machine learning algorithms, hyper-parameter selection, model ocean, and classification modules. Its web-based platform is based on Python and JavaScript. *CyberDefense* architecture shown in Fig. 1 integrates *BGPGuard* [9] and newly developed modules. It enables real-time anomaly detection and off-line classification based on machine learning algorithms.

*BGPGuard* has been used for real-time anomaly detection and off-line data classification based on BGP messages that are collected from RIPE [20] and Route Views [21] collection sites. BGP *update* messages used for real-time anomaly detection are retrieved, processed, and analyzed using developed pre-trained models. The off-line classification is based on the specified start and end data collection dates, times of the anomalous event, partitioning of the training and test datasets, and implemented machine learning algorithms: LSTM, GRU, GBDT, VFBLS, and VCFBLS [5], [6].

*CyberDefense* enables processing datasets based on connection and flow records such as NSL-KDD and CIC [22] to create models for intrusion attacks including DDoS, user to root, remote to local, and probing. It has an additional module (data container) for custom datasets.

### A. CyberDefense Modules

*Data Container*: The container enables downloading datasets collected from data communication networks. Reliable testing and validation of anomaly and intrusion detection algorithms depend on the quality of datasets including traffic collected from deployed networks or experimental testbeds. We also consider the NSL-KDD [22] dataset as one of the most widely used benchmark datasets [2], [3]. It is an improved version of the Knowledge Discovery in Databases Cup 1999 (KDD'99) dataset with removed redundancies and duplicate records. Other datasets include CIC (ISCX-IDS2012, CICIDS2017, CSE-CIC-IDS2018, CICIDS2019, CIC-Darknet2020) datasets [22]. Various custom datasets may also be included.

*Real-Time Data Retrieval*: The module is used to retrieve the latest BGP *update* message from RIPE or Route Views remote route collector (rrc) sites in 5 and 15 minutes intervals, respectively. The *time_tracker* function generates the date and time (year, month, day, hour, minute) to match the latest *update* messages downloaded from rrc sites.

*Off-Line Data Download*: This module facilitates selecting and downloading custom and BGP datasets from the *Data container*. BGP *update* messages are downloaded from the RIPE and Route Views sites. It enables locating multiple *update* messages. BGP *update* messages are downloaded based on the day of the attack, 2 days prior, and 2 days after the attack. They are initially collected in MRT format and are converted to ASCII format by using the *zebra-dump-parser* tool written in Perl. *Update* messages use the GMT time in order to synchronize RIPE and Route Views collection times.

*Feature Extraction*: Feature extraction may be customized for real-time and off-line use. The module relies on the BGP C# tool to extract 37 features from BGP *update* messages. These features capture differences between regular and anomalous data points collected from various anomalous events such as Internet worms, viruses, power outages, and ransomware attacks.

*Label Refinement*: Data are labeled based on the collected time intervals. However, regular data points may appear within the considered window of anomalous events. Hence, we refine labeling of anomalous data points by applying k-means and isolation forest (iForest) clustering algorithms. Label refinement may more accurately identify anomalous data points and, thus, improve model performance. Majority of data points within each window retain the original label (anomalies) after the label refinement. During the label refinement process, parameters of the two algorithms are tuned based on the highest silhouette coefficient. Labels of data points within the window are then updated for the subsequent data partition.

*Data Partitioning*: This module is used to create the training and test datasets based on the percentages of anomalous data. BGP datasets are created based on well-known BGP anomalies: Slammer, Nimda, and Code Red, which occurred in January 2003, September 2001, and July 2001, respectively.

*Data Processing*: The module consists of feature selection, dimension reduction, and normalization steps. Feature selection and dimension reduction rely on supervised and unsupervised learning, respectively. Feature selection and dimension reduction algorithms are useful in case of large labeled and unlabeled training data, respectively. Labeling data, usually performed by resident experts, is time-consuming. We use the extremely randomized trees (extra-trees) algorithm for feature selection. As a variant of random forests, it is used to rank features based on *Gini importance*. Variants of the autoencoder such as deep, sparse, denoising, and variational autoencoders have been used for dimension reduction. The decoder is removed after training while the output of the encoder is used as the input to gradient boosting model. We normalize datasets with $mean = 0$ and $standard\ deviation = 1$ by employing the *zscore* function.

*ML Algorithms*: This module offers options to implement deep learning and fast machine learning algorithms: CNNs, RNNs (LSTM, GRU), GBDT (XGBoost, LightGBM, CatBoost), BLS (with and without incremental learning, with cascades, VFBLS, VCFBLS).

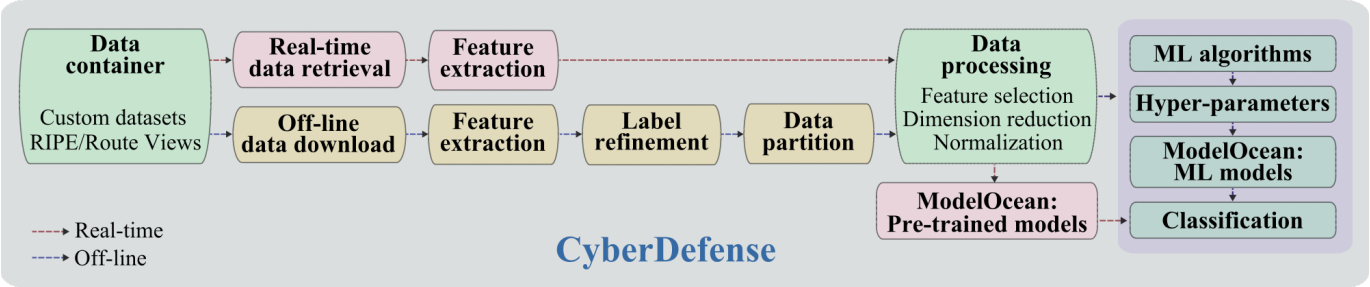


Fig. 1. Shown are modules of the *CyberDefense*: It is used to generate classification models based on various datasets. Models based on BGP *update* messages, connection and flow records, and custom datasets are created using *ModelOcean*.

Neural networks have been applied in computer vision, natural language processing, and time-series prediction including detection of network anomalies. Deep learning neural networks exhibit more robust behavior when dealing with large datasets compared to conventional machine learning algorithms. CNNs and RNNs learn the characteristics by extracting features and memorizing information, respectively.

XGBoost employs the second-order Taylor series to approximate its objective function. LightGBM is an optimized version of the traditional GBDT algorithm. It employs a histogram-based algorithm, gradient-based one-side sampling (GOSS), and exclusive feature bundling (EFB) techniques to significantly reduce the training time. CatBoost was proposed to address categorical features. VFBLS and VCFBLS have an integrated feature selection algorithm and include additional mapped features and variable number of groups of mapped features. Features are calculated and various number of relevant features is extracted from the input data. A smaller number of mapped features and groups of mapped features may be then generated leading to reduced training time and memory consumption while having performance comparable to BLS. These models offer additional flexibility and generalization.

*Selection of Hyper-Parameters*: The module is used to store the hyper-parameters leading to the best accuracy calculated during cross-validation. RNN hyper-parameters include number of hidden layers, number of hidden nodes, number of epochs, learning and dropout rates, activation function, and the optimizer. Selected hyper-parameters for GBDT models are number of boosted trees, learning rate, maximum tree depth, and number of tree leaves, and regularization terms. Performance of BLS, VFBLS, and VCFBLS models depends on the number of mapped features, groups of mapped features, and enhancement nodes. The models require additional memory and longer training time as the number of hyper-parameters increases.

*ModelOcean*: The module contains machine learning models (in *.npz* or *.pkl* formats) generated using the training datasets. The developed pre-trained models contain various deep learning RNN, GBDT, and VFBLS models used for real-time detection.

*Classification Performance*: Classification algorithms are compared based on performance metrics including training time, accuracy, F-Score (F1-Score), precision, sensitivity (recall), and elements of the confusion matrix (true positive, false positive, true negative, and false negative).

### B. CyberDefense Implementation

*CyberDefense* can be executed on the laptop, PC, and low-power devices such as Raspberry Pi. It has been tested on platforms: macOS 12.6 with 16 GB memory and Apple M1 Pro chip; Ubuntu 18.04 with 16 GB memory, Intel Core i7 7700 CPU, and GeForce GTX 1060 GPU; Raspberry Pi OS with 8 GB memory and 1.5 GHz quad-core CPU.

The *CyberDefense* offers an interactive interface for monitoring and performing experiments. Its front-end is based on HTML, cascading style sheets (CSS) (using an open-source framework *Bootstrap* and *Socket.IO* (a transport protocol written in JavaScript for real-time web applications). Its back-end is developed using *Flask* (a Python-based micro web framework). The *CyberDefense* tool relies on external libraries: CSS, JavaScript, and Python. The Python libraries installed by *pip* include *Flask* (a web framework based on *Werkzeug*) and *Jinja*.

## III. Experiments and Performance Evaluation

The *CyberDefense* tool is used to detect BGP anomalies in real-time. It also used for off-line classification of BGP anomalies caused by the recent Pakistan power outage as well as WannaCrypt and WestRock ransomware attacks.

### A. Real-Time Detection

The BGP routing traffic monitored at the rrc04 (Geneva) collection point over a 300-minute interval on June 27, 2023 is shown in Fig. 2. There were no reported anomalous events during that time. However, a small fraction of BGP routing records is predicted as anomalous because the historical BGP data may not contain sufficient information for the machine learning models to correctly perform classification.

### B. Off-Line Classification

*CyberDefense* was used for two-way classifications of BGP anomalies in case of Slammer worm, WannaCrypt ransomware, and Moscow blackout based on LSTM and GRU models with variable number of hidden layers [23]. The BGP *update* messages collected by Route Views may generate better models than those by RIPE. RNN models with 2 or 3 hidden layers outperform the models with 4

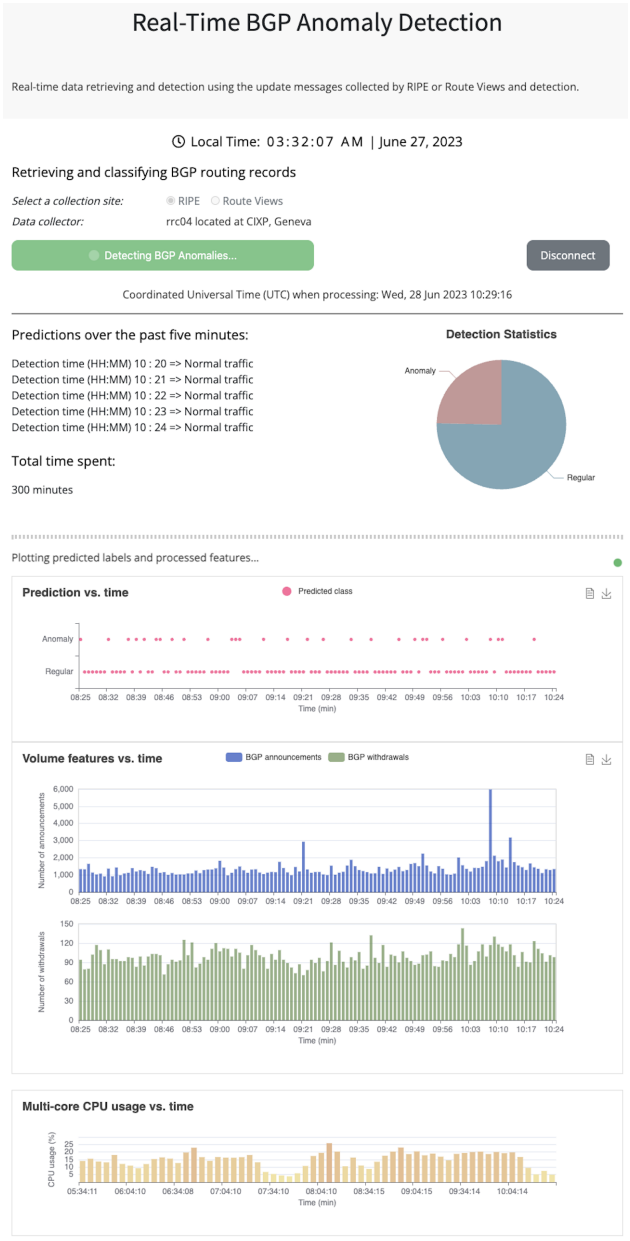


Fig. 2. The real-time detection web interface of the *CyberDefense* tool: Displayed by the front-end are prediction results over time. Plots of BGP features and CPU usage are generated by the back-end modules (real-time data retrieval, feature extraction, data processing, pre-trained models, and classification results).

layers because increasing the number of hidden layers may result in over-fitting.

*1) Power Outages:* We consider the Pakistan power outage that occurred on January 09, 2021 between 18:40 and 23:59 UTC. It was caused by a cascading effect after an abrupt frequency drop in the power transmission system of the Guddo Thermal Power Plant. The network connectivity levels in Pakistan decreased to 62 % within the first hour and to 52 % after six hours [24]. Critical infrastructure and telecommunication providers were the least affected because they relied on power backup systems. Power to most cities was restored by 12:00 UTC on January 10, 2021.

BGP *update* messages were downloaded from RIPE (rrc06, Japan) and Route Views (WIDE, Japan) data collection sites located closest to the anomalous event in Pakistan. BGP *update* messages were collected over 5 days during the Pakistan power outage. Generated datasets consist of 7,200 data points with Pakistan power outage lasting 320 min.

Classification is performed using CNN, RNNs (LSTM, GRU), bidirectional RNNs (Bi-LSTM, Bi-GRU), and LightGBM. The CNN model consists of 2 convolutional, 2 maximum pooling, and 3 fully-connected layers. Feature maps are generated using 24 and 12 kernels in the first and second convolutional layers, respectively. RNN and Bi-RNN deep neural network models consist of 1 RNN and 1 Bi-RNN layer, respectively, and up to 3 fully-connected (FC) layers. The first layer consists of 37 RNNs or Bi-RNNs. The hidden layers consist of 64 ($FC_1$), 32 ($FC_2$), and 16 ($FC_3$) nodes, respectively. The "Adam" algorithm is used to optimize RNN/Bi-RNN models using 50 epochs. The best LightGBM performance is achieved using 50 estimators and learning rate 0.1.

The best performance using CNN, LSTM, GRU, and LightGBM models is shown in Table I. The highest accuracy is achieved when using the Bi-$GRU_3$ model with the k-means label refinement. The $LSTM_4$ model with the iForest label refinement leads to the highest F-Score and precision. Telecommunication providers were the least affected during the power outage and, hence, BGP records do not capture characteristics of the event thus leading to low F-Score.

*2) Ransomware Attacks:* Recent ransomware attacks that used sophisticated cryptography techniques have significantly affected numerous organizations. These ransomware attacks are categorized into cryptoworm, ransomware-as-a-service, and automated active adversary [23]. We consider WannaCrypt and WestRock ransomware attacks. WannaCrypt targeted Windows 7 systems through the EternalBlue exploit and DoublePulsar backdoor. The WannaCrypt ransomware event lasted between 00:00 UTC on 12.05.2017 and 23:59 UTC on 15.05.2017. The malware encrypted files with 128-bit advanced encryption and demanded ransom for decryption. WannaCrypt spread rapidly by exploiting the server message block protocol vulnerabilities in Windows. It changed the desktop background to a message demanding ransom. The ransomware checked for connections to two specific domains and would stop replicating if these domains appeared registered. The WestRock ransomware attack in early 2021 targeted WestRock, the second largest packaging company in USA. It impacted the company's information (IT) and operational (OT) technology systems over six days. The WestRock ransomware attack lasted between 1:12 UTC on 23.01.2021 and 23:59 UTC on 29.01.2021.

We compare classification using BLS, VFBLS, VCFBLS, XGBoost, LightGBM, and CatBoost models. The best parameters for the developed models are listed in Table II.

The best model performance is shown in Tables III and IV. The incremental VFBLS and VCFBLS models achieve better performance with shorter training time compared to

TABLE I

PAKISTAN POWER OUTAGE: THE BEST PERFORMANCE OF CNN, LSTM, GRU, BI-LSTM, BI-GRU, AND LIGHTGBM MODELS

| Model | Refinement | Collection site | Training time (s) | Accuracy (%) | F-Score (%) | Precision (%) | Sensitivity (%) | TP | FP | TN | FN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CNN | none | RIPE | 51.00 | 84.93 | 7.00 | 4.64 | 14.17 | 17 | 349 | 2,531 | 103 |
| | | Route Views | 52.01 | 95.00 | 3.82 | 8.11 | 2.50 | 3 | 34 | 2,846 | 117 |
| | k-means | RIPE | 50.99 | 93.50 | 4.88 | 5.62 | 4.31 | 5 | 84 | 2,800 | 111 |
| | | Route Views | 52.00 | 95.87 | 1.59 | 12.50 | 0.85 | 1 | 7 | 2,875 | 117 |
| | iForest | RIPE | 50.81 | 86.53 | 8.18 | 5.63 | 15.00 | 18 | 302 | 2,578 | 102 |
| | | Route Views | 57.16 | 83.37 | 6.03 | 3.89 | 13.33 | 16 | 395 | 2,485 | 104 |
| $LSTM_4$ | none | RIPE | 45.05 | 92.83 | 4.44 | 4.76 | 4.17 | 5 | 100 | 2,780 | 115 |
| | | Route Views | 42.29 | 95.77 | 14.77 | 37.93 | 9.17 | 11 | 18 | 2,862 | 109 |
| $LSTM_2$ | k-means | RIPE | 32.42 | 93.93 | 7.14 | 8.75 | 6.03 | 7 | 73 | 2,811 | 109 |
| | | Route Views | 32.15 | 95.70 | 12.24 | 31.03 | 7.63 | 9 | 20 | 2,862 | 109 |
| $GRU_3$ | iForest | RIPE | 66.47 | 93.03 | 3.69 | 4.12 | 3.33 | 4 | 93 | 2,787 | 116 |
| $LSTM_4$ | | Route Views | 41.93 | 95.83 | 14.97 | 40.74 | 9.17 | 11 | 16 | 2,864 | 109 |
| $Bi\text{-}LSTM_2$ | none | RIPE | 25.83 | 95.57 | 9.52 | 25.93 | 5.83 | 7 | 20 | 2,860 | 113 |
| $Bi\text{-}GRU_2$ | | Route Views | 41.92 | 95.60 | 2.94 | 12.50 | 1.67 | 2 | 14 | 2,866 | 118 |
| $Bi\text{-}LSTM_3$ | k-means | RIPE | 29.94 | 95.57 | 11.92 | 25.71 | 7.76 | 9 | 26 | 2,858 | 107 |
| $Bi\text{-}LSTM_2$ | | Route Views | 43.37 | 95.73 | 3.03 | 14.29 | 1.69 | 2 | 12 | 2,870 | 116 |
| $Bi\text{-}GRU_3$ | iForest | RIPE | 27.71 | 95.90 | 8.89 | 40.00 | 5.00 | 6 | 9 | 2,871 | 114 |
| $Bi\text{-}LSTM_2$ | | Route Views | 43.40 | 95.77 | 3.05 | 18.18 | 1.67 | 2 | 9 | 2,871 | 118 |
| LightGBM | none | RIPE | 0.04 | 95.87 | 3.13 | 25.00 | 1.60 | 2 | 6 | 2,874 | 118 |
| | | Route Views | 0.05 | 94.30 | 5.59 | 8.47 | 4.17 | 5 | 54 | 2,826 | 115 |
| | k-means | RIPE | 0.01 | 93.00 | 7.08 | 7.27 | 6.90 | 8 | 102 | 2,782 | 108 |
| | | Route Views | 0.11 | 93.77 | 6.97 | 8.43 | 5.93 | 7 | 76 | 2,806 | 111 |
| | iForest | RIPE | 0.01 | 94.33 | 6.59 | 9.68 | 5.00 | 6 | 56 | 2,824 | 114 |
| | | Route Views | 0.04 | 91.90 | 6.90 | 6.38 | 7.50 | 9 | 132 | 2,748 | 111 |

TABLE II

THE BEST MODEL PARAMETERS: BLS, VFBLS, VCFBLS, XGBOOST, LIGHTGBM, AND CATBOOST

| | |
|---|---|
| | **Incr. RBF-BLS, Incr. CEBLS** |
| Incremental learning steps | 2 (RIPE, Route Views) |
| Data points/step | WannaCrypt: 1,260 (RIPE), 840 (Route Views) WestRock: 1,972 (RIPE), 1,195 (Route Views) |
| Enhancement nodes/step | 20 (RIPE), 40 (Route Views) |
| | **Incr. VFBLS, Incr. VCFBLS** |
| Incremental learning steps | 2 (RIPE, Route Views) |
| Data points/step | WannaCrypt: 315 (RIPE), 210 (Route Views) WestRock: 448 (RIPE), 229 (Route Views) |
| Feature weight for initial step | 0.9 (RIPE, Route Views) |
| Enhancement nodes/step | 20 (RIPE, Route Views) |
| | **XGBoost, LightGBM, CatBoost** |
| Number of estimators | 300, 300, 200 |
| Learning rate | 0.1 (none)/0.01 (iForest), 0.05, 0.05 |

incremental BLS models for WannaCrypt and WestRock datasets. Incremental BLS, VFBLS, and VCFBLS models require longer training time than non-incremental models because of a broader structure for mapped features and enhancement nodes needed to generate the output weights. VFBLS and VCFBLS models often require shorter training time than BLS models because they employ variable number of mapped features and selected feature selection algorithms. Incremental BLS with radial basis function (RBF-BLS) and cascades with enhancement nodes (CEBLS) lead to higher sensitivity than GBDT models. The sensitivity of models is higher for the WestRock ransomware dataset. GBDT models have the shortest training time due to the smaller number of estimators. In the WannaCrypt case, LightGBM models exhibit higher accuracy and F-Score.

Prior reported results [6], [8], [9], [23] included performance evaluation of machine learning algorithms such as CNN, RNN, Bi-RNN, GBDT, and BLS for detecting network anomalies using datasets collected from deployed networks (RIPE and Route Views) [20], [21] and testbeds (CIC) [22]. In this study, we develop new models based on data collected during the Pakistan power outage [24] and the WannaCrypt and WestRock ransomware attacks. The RNNs and Bi-RNNs models outperformed CNN models in most cases because RNNs and Bi-RNNs are designed to process sequential data. The BLS and GBDT models achieved comparable performance using larger datasets (NSL-KDD and CIC). Performance of models based on synthetically generated datasets (NSL-KDD, CICIDS2017, CSE-CIC-IDS2018, CICDDoS2019) was often higher than using data from deployed networks. The LightGBM models required the shortest training time.

## IV. CONCLUSION

Machine learning models have been compared using datasets collected during the Pakistan power outage as well as WannaCrypt and WestRock ransomware attacks. The difference in model performance is attributed to the nature of the anomalous events and the unique characteristics of each dataset. We presented the *CyberDefense* tool for detecting various network anomalies using deep learning and fast machine learning algorithms. This modular tool enables real-time and off-line detection of anomalies based on routing records downloaded from RIPE and Route Views collection sites. Custom datasets for network intrusion detection may also be used for generating new models.

TABLE III

WANNACRYPT RANSOMWARE ATTACK: THE BEST PERFORMANCE OF BLS, VFBLS, VCFBLS, XGBOOST, LIGHTGBM, AND CATBOOST MODELS

| Model | Refinement | Collection site | Training time (s) | Accuracy (%) | F-Score (%) | Precision (%) | Sensitivity (%) | TP | FP | TN | FN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RBF-BLS | none | RIPE | 3.67 | 55.73 | 56.68 | 50.48 | 64.62 | 1,512 | 1,483 | 1,397 | 828 |
| Incr. CEBLS | | Route Views | 16.73 | 56.65 | 63.97 | 50.98 | 85.85 | 2,099 | 1,932 | 948 | 33 |
| RBF-BLS | iForest | RIPE | 1.02 | 55.61 | 56.46 | 50.37 | 64.22 | 1,502 | 1,480 | 1,401 | 837 |
| Incr. CEBLS | | Route Views | 14.81 | 56.82 | 60.98 | 51.24 | 75.29 | 1,761 | 1,676 | 1,205 | 578 |
| VFBLS | none | RIPE | 6.49 | 55.06 | 46.07 | 49.85 | 42.82 | 1002 | 1008 | 1872 | 1338 |
| Incr. VFBLS | | Route Views | 4.86 | 56.82 | 64.10 | 51.10 | 85.98 | 2,012 | 1,926 | 954 | 328 |
| VFBLS | iForest | RIPE | 6.36 | 55.04 | 46.06 | 49.80 | 42.84 | 1,002 | 1,010 | 1,871 | 1,337 |
| Incr. VFBLS | | Route Views | 4.83 | 57.09 | 64.10 | 51.27 | 85.46 | 1,999 | 1,900 | 981 | 340 |
| CatBoost | none | RIPE | 1.09 | 60.31 | 62.04 | 54.30 | 72.35 | 1,693 | 1,425 | 1,455 | 647 |
| XGBoost | | Route Views | 0.87 | 53.05 | 59.56 | 48.51 | 77.14 | 1,805 | 1,916 | 964 | 535 |
| LightGBM | iForest | RIPE | 0.15 | 66.08 | 61.41 | 54.17 | 70.88 | 1,658 | 1,403 | 1,478 | 681 |
| | | Route Views | 0.23 | 52.38 | 58.95 | 48.02 | 76.31 | 1,785 | 1,932 | 949 | 554 |

TABLE IV

WESTROCK RANSOMWARE ATTACK: THE BEST PERFORMANCE OF BLS, VFBLS, VCFBLS, XGBOOST, LIGHTGBM, AND CATBOOST MODELS

| Model | Refinement | Collection site | Training time (s) | Accuracy (%) | F-Score (%) | Precision (%) | Sensitivity (%) | TP | FP | TN | FN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Incr. RBF-BLS | none | RIPE | 1.71 | 58.20 | 73.55 | 58.18 | 99.98 | 3,999 | 2,875 | 5 | 1 |
| Incr. CEBLS | | Route Views | 23.33 | 57.89 | 73.31 | 58.05 | 99.48 | 3,979 | 2,876 | 4 | 21 |
| Incr. RBF-BLS | iForest | RIPE | 33.28 | 58.20 | 73.54 | 58.16 | 99.98 | 3,998 | 2,876 | 5 | 1 |
| | | Route Views | 7.01 | 58.15 | 73.52 | 58.16 | 99.93 | 3,997 | 2,876 | 4 | 3 |
| Incr. VCFBLS | none | RIPE | 12.04 | 58.23 | 73.57 | 58.19 | 99.98 | 3,999 | 2,873 | 7 | 1 |
| | | Route Views | 9.08 | 58.30 | 73.57 | 58.25 | 99.85 | 3,994 | 2,863 | 17 | 6 |
| Incr. VFBLS | iForest | RIPE | 11.60 | 58.27 | 73.55 | 58.23 | 99.80 | 3,991 | 2,863 | 18 | 8 |
| | | Route Views | 7.62 | 58.20 | 73.55 | 58.18 | 99.98 | 3,999 | 2,875 | 5 | 1 |
| XGBoost | none | RIPE | 0.54 | 60.44 | 73.38 | 60.26 | 93.80 | 3,752 | 2,474 | 406 | 248 |
| CatBoost | | Route Views | 0.31 | 58.17 | 73.53 | 58.16 | 99.95 | 3,998 | 2,876 | 4 | 2 |
| XGBoost | iForest | RIPE | 0.52 | 59.84 | 73.05 | 59.88 | 93.62 | 3,744 | 2,508 | 373 | 255 |
| CatBoost | | Route Views | 0.48 | 58.24 | 73.53 | 58.22 | 99.78 | 3,991 | 2,864 | 16 | 9 |

## REFERENCES


[1] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 686–728, First quarter 2019.

[2] L. N. Tidjon, M. Frappier, and A. Mammar, "Intrusion detection systems: a cross-domain overview," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3639–3681, Fourth quarter 2019.

[3] B. Molina-Coronado, U. Mori, A. Mendiburu, and J. Miguel-Alonso, "Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2451–2479, Dec. 2020.

[4] C. L. P. Chen, Z. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1191–1204, Apr. 2019.

[5] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*. Cambridge, MA, USA: The MIT Press, 2022.

[6] Z. Li, A. L. Gonzalez Rios, and Lj. Trajković, "Machine learning for detecting anomalies and intrusions in communication networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2254–2264, July 2021.

[7] D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, and X. Yin, "Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2632–2647, Aug. 2021.

[8] Z. Li, A. L. Gonzalez Rios, and Lj. Trajković, "Classifying denial of service attacks using fast machine learning algorithms," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Melbourne, Australia, Oct. 2021, pp. 1221–1226.

[9] ——, "Machine learning for detecting the WestRock ransomware attack using BGP routing records," *IEEE Commun. Mag.*, vol. 61, no. 3, pp. 20–26, Mar. 2023.

[10] (2023, June) Snort - Network Intrusion Detection & Prevention System. [Online]. Available: https://www.snort.org.

[11] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: an intelligent anomaly-based intrusion detection system for IoT edge devices," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6882–6897, Aug. 2020.

[12] P. Mishra, V. Varadharajan, E. S. Pilli, and U. Tupakula, "VMGuard: a VMI-based security architecture for intrusion detection in cloud environment," *IEEE Trans. Cloud Comput.*, vol. 8, no. 3, pp. 957–971, July–Sept. 2020.

[13] D. Jin, Y. Lu, J. Qin, Z. Cheng, and Z. Mao, "SwiftIDS: real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism," *Computers & Security*, vol. 97, no. 101984, pp. 1–12, Oct. 2020.

[14] Z. A. E. Houda, L. Khoukhi, and A. S. Hafid, "Bringing intelligence to software defined networks: mitigating DDoS attacks," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2523–2535, Dec. 2020.

[15] (2023, June) BGProtect. [Online]. Available: https://www.bgprotect.com.

[16] (2023, June) Secure IPS (NGIPS). [Online]. Available: https://www.cisco.com/c/en_ca/products/security/ngips/index.html.

[17] (2023, June) FortiGuard IPS Security Service. [Online]. Available: https://www.fortinet.com/products/ips.

[18] (2023, June) Advanced Threat Prevention. [Online]. Available: https://www.paloaltonetworks.com/network-security/advanced-threat-prevention.

[19] (2023, June) CyberDefense: Tool for Detecting Network Anomalies and Intrusions. [Online]. Available: https://github.com/zhida-li/cyberDefense.

[20] (2023, June) RIPE Network Coordination Centre. [Online]. Available: https://www.ripe.net.

[21] (2023, June) University of Oregon Route Views project. [Online]. Available: http://www.routeviews.org.

[22] (2023, June) Canadian Institute for Cybersecurity datasets. [Online]. Available: https://www.unb.ca/cic/datasets/index.html.

[23] Z. Li, A. L. Gonzalez Rios, and Lj. Trajković, "Detecting Internet worms, ransomware, and blackouts using recurrent neural networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Toronto, Canada, Oct. 2020, pp. 2165–2172.

[24] (2023, June) Pakistan Internet connectivity collapses amid nation-scale power outage. [Online]. Available: https://bit.ly/3ICKQeZ.