RESEARCH ARTICLE WWW.PEGEGOG.NET

Dynamic Defence: Safeguarding Software Defined Network Against Botnet Threats

[1]DR.A.NAGARJUNA REDDY, [2]SABAH NAAZ SAHERWARDI, [3]SYEDA MUNEEBA MAHNAZ, [4]D. MAHESHWARI

[1]Professor Department of Computer Science and Engineering, Sridevi Women's Engineering College, Hyderabad, India.

Email: swecanr@gmail.com,

[2,3,4,]B.Tech Student, Department of Computer Science and Engineering, Sridevi Women's Engineering College, Hyderabad, India.

Abstract:

One new design that makes managing and communicating across large-scale networks easier and more flexible is software-defined networking, or SDN. It allows for the smooth and dynamic execution of complicated network choices via programmable and centralised interfaces. But with SDN, companies and people may create network apps that meet their needs and enhance their services. On the other side, it opened itself up to a host of new privacy and security risks while also introducing the possibility of a single point of failure. Attackers often use OpenFlow switches to launch botnets and Distributed Denial of Service (DDoS) assaults on the controller. Popular security apps that use deep learning (DL) to quickly identify and counteract attacks are on the rise. Here, we examine botnet-based DDoS attack detection using DL approaches in an SDN-supported context and demonstrate their performance. For the assessment, we utilise a dataset that we personally created recently. In order to choose the most useful subset of characteristics, we used feature weighting and tuning techniques. Using both a synthetic dataset and actual testbed conditions, we validate the measurements and simulation results. To identify botnet-based DDoS assaults using easily-obtained characteristics and data, this work primarily seeks to identify a lightweight DL approach with baseline hyper-parameters. We found that the DL technique's performance is affected by the optimal subset of features, and that the prediction accuracy of the same approach may be varied with a different collection of features. Lastly, we discovered that the CNN approach works better than both the dataset and actual testbed settings based on empirical findings. For typical flows, CNN achieves a detection rate of 99%; for malicious flows, it drops to 97%.

INTRODUCTION

We have reached the limits of conventional networks, yet the internet is expanding at a dizzying rate. Patching the network helps fix new problems with traditional networks, but it also makes the network larger and less controllable. These issues have been addressed with the creation of Software-

Corresponding Author e-mail: swecanr@gmail.com
How to cite this article: 1DR.A.NAGARJUNA REDDY, 2SABAH NAAZ SAHERWARDI, 3SYEDA MUNEEBA MAHNAZ, 4D. MAHESHWARI. Dynamic Defence: Safeguarding Software Defined Network Against Botnet Threats.Pegem Journal of Education and Instruction, Vol. 13, No. 4, 2023, 407-417.**Source of support:** Nil **Conflicts of Interest:** None.

DOI: 10.48047/pegegog.13.04.48

Received: 12.10.2023

Accepted: 22.11.2023 **Published:** 24.12.2023

Defined Networking (SDN), which separates the data plane from the control plane. SDN rose to prominence among the This paper was reviewed and approved for publication by Cheng Chin, an associate editor. The network community is interested in it because of its innovative design and its ability to meet the needs of rapidly expanding networks. Thanks to SDN's centralised control architecture, controllers may manage the whole network using open south API interfaces and access any nearby OpenFlow switches. The application, control, and data layers make up the three-layer network architecture. All of the rules and policies set by the network administrator are executed at the application layer, and these rules and policies may be dynamically adopted by the SDN controller. The network's behaviour is susceptible to changes made to the application layer. The open-source platform's application layer is a great improvement, VOLUME 11, 2023 49153. The administrator is not compelled to depend only on suppliers, according to IEEE Transaction on Machine Learning, Volume:11, Issue Date:17.May. 2023. One positive aspect of SDN is that it enables administrators to build specialised network apps in the cloud using general-purpose hardware, without worrying about licence limits. Running in the control layer—sometimes called the "brain" of the architecture—are SDN controllers. Controllers take rules sent by the application layer, convert them into human-readable messages, and send them on to the data layer. The data layer then provides feedback, which the controllers then relay to the application layer. In addition, the control layer decides and the data layer implements the rules. The data layer receives instructions from the control layer and contains various hardware devices like routers and OpenFlow switches; nevertheless, it does not possess intelligence of its own. In addition, it streamlines and makes development, implementation, and maintenance of the network easier. It is simple to update and add new programmes that improve the network's capabilities and features. In addition to being inexpensive, SDN-based networks are compatible with almost all devices and need just minimal hardware. Access to the network is granted without disclosing the specifics of the many underlying levels. While software-defined networking (SDN) is a fantastic innovation that has the potential to make networks more adaptable and controllable, it also has the potential drawbacks of being too centralised, making it easy for a single person to oversee the whole system. While SDN does assist with the security of older networks, it is still not a reliable enough security solution to back the next generation of networking. New security risks and the possibility of a single point of failure may result from its novel design and centralised control nature. In addition, as a result of SDN's centralization, attackers are able to conduct several forms of assaults, including botnets, DDoS, saturation, and more. In the future generation of networks, botnet attacks—which are harmful—will pose a significant hazard. Botnets are hostile networks that use compromised computers to conduct attacks including distributed denial of service (DDoS), identity theft, spamming, phishing, and domain name system spoofing. To gain control of a single device without affecting its legitimate users, a malevolent actor known as a "bot master" attempts to get unauthorised access to the device and then uses botnet malware. The next step is to link the bots to the attacker's Command and Control (C&C) centre; once connected, the bots will wait to carry out harmful operations as instructed. The most advanced distributed denial of service (DDoS) attacks in SDN and IoT networks nowadays usually use botnet technologies. Botnet technology can launch several forms of distributed denial of service attacks due to its robustness and adaptability.

RELATED WORK

Classification of potential dangers to networks and how existing data sets impact intrusion detection systems

Secure application, system, and network development is one of the most pressing issues of the present decade, given the world's growing reliance on computers and automation. Due to the ever-increasing complexity of today's networks and services, the number of dangers that people and companies confront is growing at an exponential rate. Researchers have suggested a plethora of anomaly detection methods to mitigate these dangers, but existing technologies aren't always up to the task of keeping up with dynamic architectures, related risks, and zero-day assaults. With the proliferation of complex threats and the limitations of existing datasets, this book seeks to identify these issues and how they affect the development of Network Intrusion Detection Systems (NIDS). A taxonomy of network threats and related tools for these attacks, as well as a survey of prominent datasets and an analysis of their usage and impact on the development of Intrusion Detection Systems (IDS) over the last decade are two essential pieces of information that this manuscript offers to researchers. The paper emphasises that only 33.3% of our threat taxonomy is covered by existing IDS research. Machine learning intrusion detection systems are currently limited in their ability to identify real-world attacks because datasets lack attack representation, real-world threats are not available, and there are many outdated threats included. Improving dataset generation and real-world data collecting is the goal of this manuscript's unique blend of taxonomy and dataset analysis. Consequently, this will make new datasets more accurate reflections of network risks and make next-generation IDS more efficient.

Machine learning data analytics in online education: problems and potential solutions

As more and more people have access to the internet, e-learning has become a hot topic. Reason being, it has opened up a world of knowledge to people all over the globe. The ever-increasing quantity of data being produced by various sensors and gadgets used around the globe is already a result of this. The need to examine gathered data and derive valuable insights from it has resulted from this. Proposed methods such as data analytics (DA) and machine learning (ML) may aid in data extraction and the discovery of useful patterns. The article delves into the definitions and characteristics of e-learning as a discipline. Additionally, the many difficulties encountered by each party involved in this procedure are addressed. Also included are some of the published publications that attempt to address these issues. Following that, a concise overview of some widely used ML and DA methods is provided. Lastly, we suggest a few study possibilities that make use of these methods in order to shed light on the areas that should be further investigated.

Cisco Projects More Internet Protocol (IP) Traffic in the Next Five Years Than the Internet Has Ever Seen

But according to Cisco's latest Visual Networking Index (VNI), that's only the start. More information packets will travel across international networks in 2022 than in the whole "internet years" up to the end of 2016. That is to say, in 2022, more traffic will be generated than in the whole 32-year history of the internet. How will that traffic be generated? Everyone, every computer, and every internet user. Half of the world's population will have access to the internet by the year 2022. The number of connected devices and users will exceed 28 billion. Additionally, 82% of all IP traffic will be video.The exponential growth in both the scope and depth of the internet is something that few could have predicted. According to Jonathan Davidson, senior vice president and general manager of Cisco's Service Provider Business, traffic has grown at a CAGR of 36 percent since the VNI Forecast was launched in 2005, with a 56-fold rise in the number of devices, apps, and users using IP networks. Improved traffic management and routing, together with the delivery of premium experiences, is the primary goal of network transformation initiatives undertaken by global service providers. We learn from our clients' experiences and pass that knowledge on to them so they can make the necessary technological and architectural shifts for success.

Big healthcare data and the proliferation of the internet of things (IoT): a literature review and areas for further study

Achieving better health is a goal of sustainable development in every country. There are many unexplored applications of the Internet of Things in this field. Prioritising the use of the Internet of Things (IoT) in healthcare in order to attain sustainable development is the goal of this study. Based on the information gathered, the study may be classified as an applied descriptive research. This study is a single-sectional survey based on FAHP technique. Next, we prioritised the use of the Internet of Things (IoT), agreed upon paired comparison matrices, and assigned them to weighted criteria. Findings indicate that "Economic Prosperity" and "Quality of Life" were the two most important factors for healthcare IoT sustainable development. In addition, "Ultraviolet Radiation," "Dental Health," and "Fall Detection" were determined to be the most important health-related IoT objectives based on use.

A smart city framework for the continuous, diverse availability of vehicular cloud services

One important and necessary part of smart city design is the intelligent and linked transport system (ICTS). Services provided by ICTS include shared multimedia content, vehicle power management, and route navigation. An ongoing challenge for smart cities is the efficient and dependable selection of services for smart cars, as they implement various technologies to enhance the variety and performance of vehicular cloud services. On top of that, SPs can only guarantee the quantity, quality, and accessibility of the services they provide to cloud subscribers in vehicles. In order to get the necessary services while in motion, smart cars depend on many SPs. Consequently, it becomes more difficult for customers to vehicle cloud services to get services that match their preferences for quality of experience (QoE). This study presents a novel service provisioning approach that uses a cluster-based trustworthy third party (TTP) infrastructure to provide cloud customers for automotive applications continuous availability of diverse cloud services. Third-party intermediaries (TTPs) mediate disputes between users and cloud service providers. Clusters are created for certain services based on the vehicles' movement patterns and other features that are thought to be comparable. With the help of cluster chiefs and service suppliers, TTPs negotiate for services that have high quality of experience features. It is possible to negotiate services prior to a vehicle's arrival by using a location prediction approach to ascertain the vehicle's future position. We provide simulated results that demonstrate our method accurately discovers and delivers cloud services with improved QoE, decreased end-to-end latency, and low overhead.

METHODOLOGY

In order to carry out this project, we have customised the following modules and used the same dataset that was provided in your requirement file.

1. Dataset upload: this is the module we'll use to upload datasets.

2. Preparing the dataset: here is where we'll divide and clean the data.

3. Execute the Default Decision Tree Algorithm: We trained the Default Decision Tree algorithm using this module and achieved an accuracy of 99.23%.

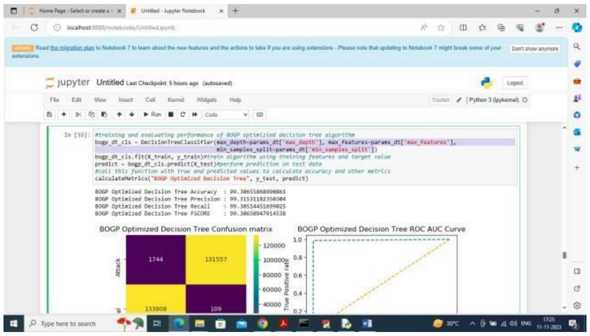
4. Execute the BOGP Optimised Decision Tree Algorithm: By training the decision tree with the best selected BOGP parameters, this module achieved a 99.30% accuracy rate, which is higher than the default decision tree.

5. Execute the SVM Algorithm: By using this module, the existing SVM algorithm was trained to achieve a 96% accuracy rate.

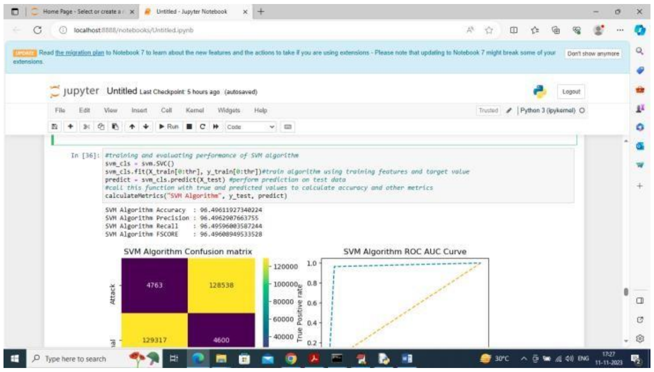
6. Execute the Convolutional Neural Network (CNN) Algorithm: An accuracy of 99.996% was achieved with the use of this module's training extension: CNN.

7. Comparison Graph: this module allows you to display a comparison of all methods. The x-axis shows the names of the algorithms, and the y-axis shows metrics like accuracy and other metrics in various colour bars. You can see that all of the algorithms demonstrate good performance when it comes to CNN extension.

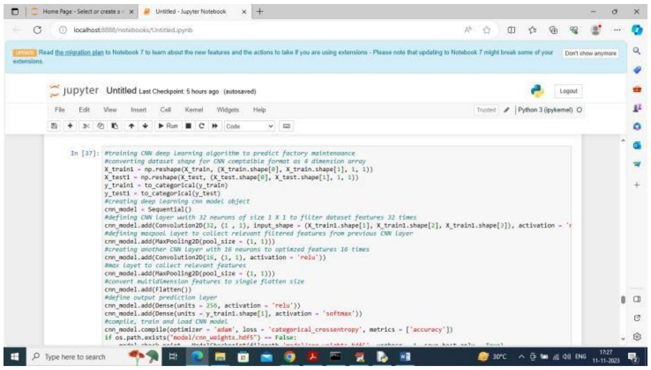
RESULT AND DISCUSSION


```
In [35]: #training and evaluating performance of BOGP optimizied decision tree algorithm
         bogp_dt_cls = DecisionTreeClassifier(max_depth=params_dt['max_depth'], max_features=params_dt['max_features'],
                                              min_samples_split=params_dt['min_samples_split'])
         bogp_dt_cls.fit(X_train, y_train)#train algorithm using training features and target value
         predict = bogp_dt_cls.predict(X_test)#perform prediction on test data
         #call this function with true and predicted values to calculate accuracy and other metrics
         calculateMetrics("BOGP Optimized Decision Tree", y_test, predict)

BOGP Optimized Decision Tree Accuracy  : 99.30655868990861
BOGP Optimized Decision Tree Precision : 99.31531182350304
BOGP Optimized Decision Tree Recall    : 99.30514451699025
BOGP Optimized Decision Tree FSCORE    : 99.30650947914538
```
BOGP Optimized Decision Tree Confusion matrix
1744
131557
133808
109
Attack
120000
100000
80000
60000
40000
BOGP Optimized Decision Tree ROC AUC Curve
True Positive rate
1.0
0.8
0.6
0.4
0.2

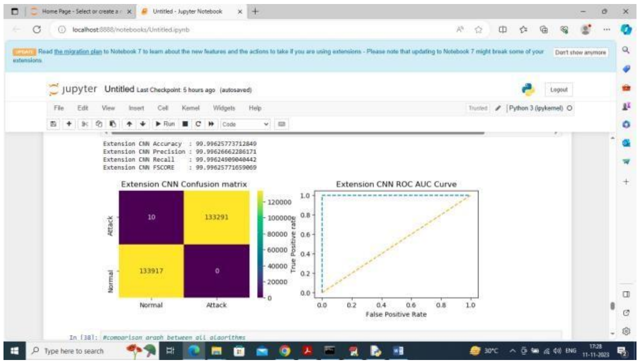
The following screen displays the results of training a decision tree using the best-selected BOGP parameters; the optimised decision tree outperformed the default decision tree with an accuracy of 99.30%; and other metrics values are also shown.


```
In [36]: #training and evaluating performance of SVM algorithm
         svm_cls = svm.SVC()
         svm_cls.fit(X_train[0:thr], y_train[0:thr])#train algorithm using training features and target value
         predict = svm_cls.predict(X_test) #perform prediction on test data
         #call this function with true and predicted values to calculate accuracy and other metrics
         calculateMetrics("SVM Algorithm", y_test, predict)

SVM Algorithm Accuracy  : 96.49611927340224
SVM Algorithm Precision : 96.4962907663755
SVM Algorithm Recall    : 96.49596003587244
SVM Algorithm FSCORE    : 96.49608949533528
```

Using the current SVM technique for training, the above screen achieved an accuracy of 96%.

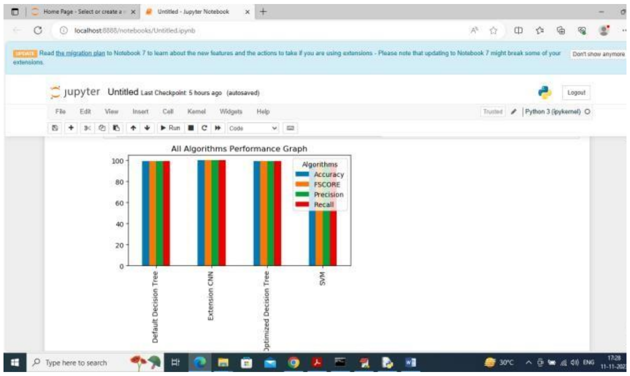

```
In [37]: #training CNN deep learning algorithm to predict factory maintenaance
         #converting dataset shape for CNN comptaible format as 4 dimension array
         X_train1 = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1, 1))
         X_test1 = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1, 1))
         y_train1 = to_categorical(y_train)
         y_test1 = to_categorical(y_test)
         #creating deep learning cnn model object
         cnn_model = Sequential()
         #defining CNN layer with 32 neurons of size 1 X 1 to filter dataset features 32 times
         cnn_model.add(Convolution2D(32, (1 , 1), input_shape = (X_train1.shape[1], X_train1.shape[2], X_train1.shape[3]), activation = 'r
         #defining maxpool layet to collect relevant filtered features from previous CNN layer
         cnn_model.add(MaxPooling2D(pool_size = (1, 1)))
         #creating another CNN layer with 16 neurons to optimzed features 16 times
         cnn_model.add(Convolution2D(16, (1, 1), activation = 'relu'))
         #max layet to collect relevant features
         cnn_model.add(MaxPooling2D(pool_size = (1, 1)))
         #convert multidimension features to single flatten size
         cnn_model.add(Flatten())
         #define output prediction layer
         cnn_model.add(Dense(units = 256, activation = 'relu'))
         cnn_model.add(Dense(units = y_train1.shape[1], activation = 'softmax'))
         #compile, train and load CNN model
         cnn_model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
         if os.path.exists("model/cnn_weights.hdf5") == False:
```

The following result will be obtained after running the aforementioned block and training the deep learning extension CNN algorithm in the above screen.


```
Extension CNN Accuracy  : 99.99625773712849
Extension CNN Precision : 99.99626662286171
Extension CNN Recall    : 99.99624909040442
Extension CNN FSCORE    : 99.99625771659069
```

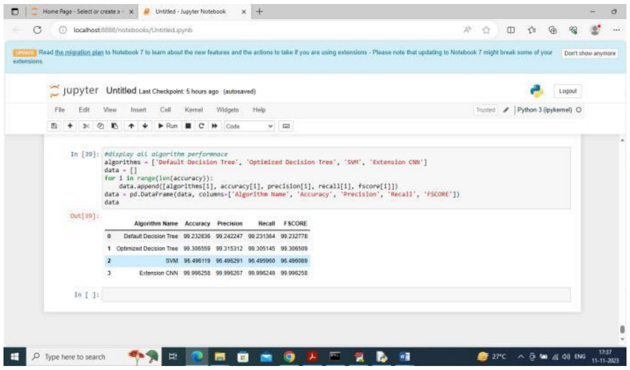
```
In [38]: #comparison graph between all algorithms
```

Its accuracy in the aforementioned screen extension was 99.996%.


Home Page - Select or create a
Untitled - Jupyter Notebook
localhost:8888/notebooks/Untitled.ipynb
jupyter Untitled Last Checkpoint: 5 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
All Algorithms Performance Graph
100
80
60
40
20
0
Algorithms
Accuracy
FSCORE
Precision
Recall
Default Decision Tree
Extension CNN
Optimized Decision Tree
SVM
Type here to search
30°C
ENG
17:28
11-11-2023

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions. Don't show anymore

A high level of performance was achieved by CNN in all of the algorithms included in the above graph, which shows algorithm names on the x-axis and several metrics on the y-axis, including accuracy.


Home Page - Select or create a
Untitled - Jupyter Notebook
localhost:8888/notebooks/Untitled.ipynb
jupyter Untitled Last Checkpoint: 5 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
27°C
ENG
17:37
11-11-2023

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions. Don't show anymore

```
In [39]: #display all algorithm performnace
         algorithms = ['Default Decision Tree', 'Optimized Decision Tree', 'SVM', 'Extension CNN']
         data = []
         for i in range(len(accuracy)):
             data.append([algorithms[i], accuracy[i], precision[i], recall[i], fscore[i]])
         data = pd.DataFrame(data, columns=['Algorithm Name', 'Accuracy', 'Precision', 'Recall', 'FSCORE'])
         data
Out[39]:
```

	Algorithm Name	Accuracy	Precision	Recall	FSCORE
0	Default Decision Tree	99.232836	99.242247	99.231364	99.232778
1	Optimized Decision Tree	99.306559	99.315312	99.305145	99.306509
2	SVM	96.496119	96.496291	96.495960	96.496089
3	Extension CNN	99.996258	99.996267	99.996249	99.996258

```
In [ ]:
```

You can see that extended CNN achieved good results across all algorithms in the tabular presentation of algorithm performance up top.

CONCLUSION

The proliferation of Internet-of-Things (IoT) devices has skyrocketed in recent years, driven by the ever-increasing need for connection and the growing dependence on the Internet. Recent predictions indicate that there will be around 28.5 billion linked gadgets by 2022, lending credence to this claim. Since there are now more possible entry points into networks, the number of attacks targeting these systems has risen. Hence, to guarantee these devices are adequately secured, efficient and effective attack detection and mitigation strategies are required. Therefore, in order to identify botnet assaults on IoT devices, this study presented an improved ML-based framework that fused the Bayesian optimisation Gaussian Process (BO-GP) with a decision tree (DT) classification model. An efficient, effective, and dynamic framework for detecting attacks on the Internet of Things was the target of this project. The accuracy, precision, recall, and F-score were all enhanced by the suggested optimised DT-based framework, according to the experimental data. For these four measures in particular, it attained 99.99%, 0.99, 1.00, and 1.00, respectively. This demonstrated the efficacy and robustness of the suggested paradigm in identifying botnet assaults in IoT settings. Numerous avenues of expansion are open to this piece of art. An obvious way to improve the normal traces situation is to utilise the whole dataset for data oversampling, which would increase the number of normal cases. Investigating time-related characteristics for any trends or behaviours that can aid in identifying botnet assaults in IoT systems is another worthwhile topic to explore.

REFERENCES

[1] Cisco, “Cisco Predicts More IP Traffic in the Next Five Years Than in the History of the Internet,” Nov. 2018.

[2] Z. Alansari, S. Soomro, M. R. Belgaum, and S. Shamshirband, “The rise of internet of things (iot) in big healthcare data: review and open research issues,” in Progress in Advanced Computing and Intelligent Engineering. Springer, 2018, pp. 675–685.

[3] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-khah, and P. Siano, “Iot-based smart cities: A survey,” in 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), 2016, pp. 1–6.

[4] I. Al Ridhawi, M. Aloqaily, B. Kantarci, Y. Jararweh, and H. T. Mouftah, “A continuous diversified vehicular cloud service availability framework for smart cities,” Computer Networks, vol. 145, pp. 207–218, 2018.

[5] Z. Doffman, “Cyberattacks on iot devices surge 300% in 2019,‘measured in billions,’report claims,” 2019.

[6] C. Crane, “20 surprising iot statistics you don’t already know,” 2019.

[7] A. Moubayed, A. Refaey, and A. Shami, “Software-defined perimeter (sdp): State of the art secure solution for modern networks,” IEEE Network, vol. 33, no. 5, pp. 226–233, Sep.- Oct. 2019.

[8] P. Kumar, A. Moubayed, A. Refaey, A. Shami, and J. Koilpillai, “Performance analysis of sdp for secure internal enterprises,” in 2019 IEEE Wireless Communications and Networking Conference (WCNC), Apr. 2019, pp. 1–6.

[9] H. Hindy, D. Brosset, E. Bayne, A. K. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, “A taxonomy of network threats and the effect of current datasets on intrusion detection systems,” IEEE Access, vol. 8, pp. 104 650–104 675, 2020.

[10] A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya, and A. Shami, “Elearning: Challenges and research opportunities using machine learning data analytics,” IEEE Access, vol. 6, pp. 39 117–39 138, 2018.

[11] A. Moubayed, M. Injadat, A. Shami, and H. Lutfiyya, “Student engagement level in an e-learning environment: Clustering using k-means,” American Journal of

Distance Education, vol. 34, no. 2, pp. 137–156, 2020.

[12] ——, "Relationship between student engagement and performance in e-learning environment using association rules," in 2018 IEEE World Engineering Education Conference (EDUNINE), 2018, pp. 1–6.

[13] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Systematic ensemble model selection approach for educational data mining," Knowledge-based Systems, vol. 200, p. 105992, Jul. 2020.

[14] ——, "Multi-split optimized bagging ensemble model selection for multiclass educational data mining," Applied Intelligence, pp. 1–23, Jul. 2020.

[15] A. Moubayed, M. Injadat, A. Shami, and H. Lutfiyya, "DNS TypoSquatting Domain Detection: A Data Analytics & Machine Learning Based Approach," in 2018 IEEE Global Communications Conference (GLOBECOM), Dec. 2018, pp. 1–7.

[16] A. Moubayed, E. Aqeeli, and A. Shami, "Ensemble-based feature selection and classification model for dns typo-squatting detection," in 2020 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Aug. 2020.

[17] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," Neurocomputing, 2020. [Online]. Available: http://www.sciencedirect.com/science/article / pii/S0925231220311693

[18] A. Moubayed, "Optimization Modeling and Machine Learning Techniques Towards Smarter Systems and Processes," Ph.D. dissertation, University of Western Ontario, Aug. 2018.

[19] M. Injadat, "Optimized Machine Learning Models Towards Intelligent Systems," Ph.D. dissertation, University of Western Ontario, Aug. 2018.

[20] L. Yang, A. Moubayed, I. Hamieh, and A. Shami, "Tree-based intelligent intrusion detection system in internet of vehicles," in 2019 IEEE Global Communications Conference (GLOBECOM), Dec 2019, pp. 1–6.

[21] M. Injadat, F. Salo, A. B. Nassif, A. Essex, and A. Shami, "Bayesian optimization with machine learning algorithms towards anomaly detection," in 2018 IEEE Global Communications Conference (GLOBECOM), Dec 2018, pp. 1–6.

[22] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, “Multi-stage optimized machine learning framework for network intrusion detection,” IEEE Transactions on Network and Service Management, pp. 1–1, Aug. 2020.

[23] F. Salo, M. Injadat, A. Moubayed, A. B. Nassif, and A. Essex, “Clustering enabled classification using ensemble feature selection for intrusion detection,” in 2019 International Conference on Computing, Networking and Communications (ICNC), 2019, pp. 276–281.

[24] M. A. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, “Scada system testbed for cybersecurity research using machine learning approach,” Future Internet, vol. 10, no. 8, p. 76, 2018.

[25] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, “Deep recurrent neural network for iot intrusion detection system,” Simulation Modelling Practice and Theory, vol. 101, p. 102031, 2020.

[26] E. Anthi, L. Williams, M. Słowinska, G. Theodorakopoulos, and P. Bur- ´ nap, “A supervised intrusion detection system for smart home iot devices,” IEEE Internet of Things Journal, vol. 6, no. 5, pp. 9042–9053, 2019.

[27] Z. Chen, Q. Yan, H. Han, S. Wang, L. Peng, L. Wang, and B. Yang, “Machine learning based mobile malware detection using highly imbalanced network traffic,” Information Sciences, vol. 433, pp. 346–364, 2018.

[28] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.

[29] F. Hu and H. Li, “A novel boundary oversampling algorithm based on neighborhood rough set model: Nrsboundary-smote,” Mathematical Problems in Engineering, vol. 2013, 2013.

[30] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, “Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset,” Future Generation Computer Systems, vol. 100, pp. 779–796, 2019