Cooperative Machine Learning Techniques for Cloud Intrusion Detection

Zina Chkirbene[1], Ridha Hamila[1], Aiman Erbad[2], Serkan Kiranyaz[1], Nasser Al-Emadi[1], Mounir Hamdi[2]
[1]College of Engineering, Qatar University
[2]Division of Information and Computing Technology,
College of Science and Engineering,
Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar.

2021 International Wireless Communications and Mobile Computing (IWCMC) | 978-1-7281-8616-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/IWCMC51323.2021.9498809

***Abstract*—Cloud computing is attracting a lot of attention in the past few years. Although, even with its wide acceptance, cloud security is still one of the most essential concerns of cloud computing. Many systems have been proposed to protect the cloud from attacks using attack signatures. Most of them may seem effective and efficient; however, there are many drawbacks such as the attack detection performance and the system maintenance. Recently, learning-based methods for security applications have been proposed for cloud anomaly detection especially with the advents of machine learning techniques. However, most researchers do not consider the attack classification which is an important parameter for proposing an appropriate countermeasure for each attack type. In this paper, we propose a new firewall model called Secure Packet Classifier (SPC) for cloud anomalies detection and classification. The proposed model is constructed based on collaborative filtering using two machine learning algorithms to gain the advantages of both learning schemes. This strategy increases the learning performance and the system's accuracy. To generate our results, a publicly available dataset is used for training and testing the performance of the proposed SPC. Our results show that the accuracy of the SPC model increases the detection accuracy by 20% compared to the existing machine learning algorithms while keeping a high attack detection rate.**

Index— Cloud security, secure packet classifier, firewalls, intrusion detection systems, machine learning techniques.

I. Introduction

The cloud computing (CC) paradigm has attracted a lot of interest from both, industry and academia. It offers multiple services including resource pooling, multi-tenancy, and elasticity[1]. These qualities promote cloud computing as one of the most important pillars for businesses and organizations [2]. Although the cloud computing paradigm raises economic efficiency, security is still one of the significant concerns in adopting the cloud computing model [3].

To resolve this security problem, researchers proposed different firewall models and rule-based security such as intrusion detection systems (IDS), which use the attack signatures for attack identification [4], [5]. However, if the signatures are not sufficiently robust in describing the attack conditions, an attacker may access the network [6]. Also, if the installed IDS system is stopped for any reason, then the attacker may exploit the time needed for the system reparation and gain a foothold in the network. Moreover, the signature of an IDS system is created and deployed by human intervention [7]. Thus, it may take hours or days to generate a new signature for an attack which can be too long when dealing with rapidly moving attacks, such as worm propagation [8], [9].

To solve this problem, systems that do not rely on human intervention were invented such as systems based on machine learning (ML) [10], [11]. In this paper, we propose a new network security system called "secure packet classifier (SPC)" for anomaly detection using supervised learning (Figure 1) techniques. The idea is to classify the attacks rather than just detecting anomalous traffic. The SPC completes a detailed comparison between multiple machine learning algorithms to select the best two models to be combined. The main criteria for selection are the time complexity and learning performance to increase the fast processing and the detection performance of the proposed SPC.

The main contributions of this paper can be summarized as follows:

1) We present a novel firewall called secure packet classifier (SPC), capable of detecting anomalies and identifying the specific types of attacks.
2) We compare multiple ML algorithms in terms of accurate detection. Then, we select the best two models to be combined for the SPC model creation.
3) We design collaborative filtering that collects the predictions of the machine learning algorithms. The packet classifications of the two models are considered and the final class having more votes will be selected by the SPC.
4) We perform a testing step for the SPC. For each packet, the predicted class of the model is compared with the correct class provided in the testing set.

The rest of the paper is organized as follows: Section II describes the proposed framework. Section II presents the experimental results and Section IV concludes the paper.

II. Secure Packet Classifier Model

A. Overall framework

The overall structure of the anomaly detection framework is composed of three major steps. In the first step, an **offline data processing** is performed for dataset partitioning and machine learning algorithms selection. This module is used to make more appropriate data training for the model creation step. The size of this training data is set to $\alpha \times T$, where T denotes the size of the original input data and $\alpha \in [0\ 1]$.

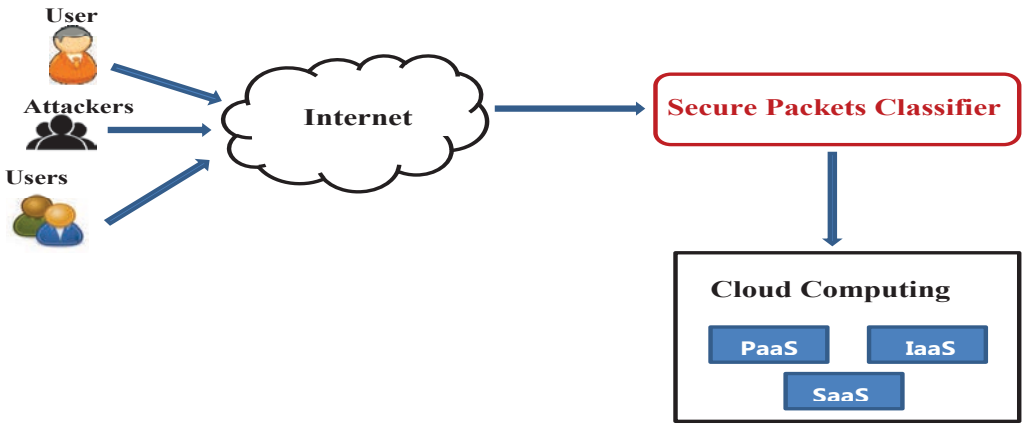

Fig. 1. The proposed model.

The second step is related to the **Model Training** which includes model creation, confusion cube construction, and SPC model generation. The packets that have been selected in the first step (αT) are pre-processed for learning algorithms inputs to create models while the full original input data (includes T packets) is used for the confusion cube construction. In fact, the proposed SPC model generation requires the confusion cube as an input to combine the two used ML techniques. Therefore, only a part of the available training input data is used for the ML model creation, then the full original input data is used to generate the confusion cube using the generated models.

The final step is **Model Testing** which entails verifying the SPC performance using a testing set and comparing the proposed SPC with real packets classification in the testing set.

B. Off-line data processing

Data processing is one of the most important steps in the proposed system because it selects the learning algorithms to be used in the SPC model. In particular, the dataset is divided in order to select the best part to be used for learning model creation. The off-line data processing is performed by the following steps:

1) Step 1: Input data for training all the potential investigated supervised learning algorithms.
2) Step 2: Select only the algorithms having low running time in order to create a combined model with low complexity.
3) Step 3: Vary the size of the training set; vary α from $[0\ 1]$ to find the best size of the training set (αT) that gives better learning performance using the selected algorithms.
4) Step 4: Analyzing the learning performance of all the possible combinations between the selected algorithms and select the best value for α and two ML algorithms to be combined having good accuracy with high precision and F1 score.

C. Train model

The training model is composed of three steps: ML model creation, confusion cube construction, and SPC model creation, as depicted in Figure 2.

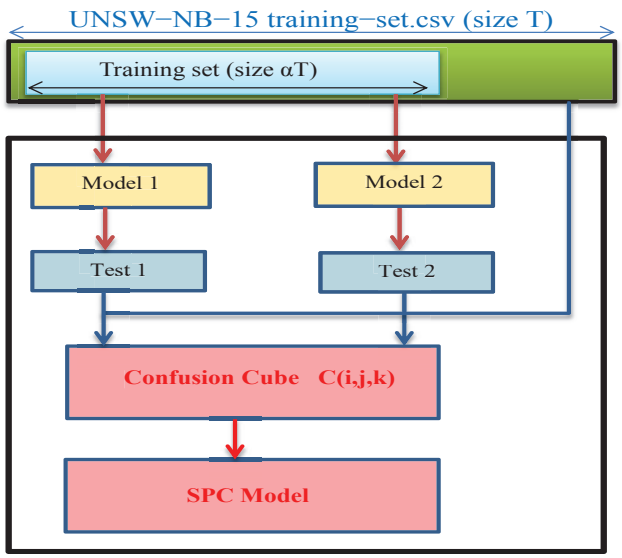

Fig. 2. Model creation using UNSW dataset.

1) Models creation:

Each selected learning algorithm takes as input the proceeded data to specify all the classifier options and create a predict function for the packets classification.

Algorithm 1 is needed to create two models that should be able to generate multiple rules for packet classifications using the two selected learning algorithms (LG_1, LG_2). Then, the created models ($Model_1$, $Model_2$) are used in the predict function $predict$ for predicting the classification of packets. The two functions $Extract\_Classes$ and $ExtractResponse$

Algorithm 1 Creation of models

Input:
LG_1: The selected learning algorithm 1
LG_2: The selected learning algorithm 2
$Data_{input}$: The input data
$Data_{prod}$: The proceeded data set.

Extract classes:
$\mathcal{S} = Extract\_Classes(Data_{prod})$

Create classifier:
$Model_1 \leftarrow ML\_Model\_create(LG1, Data_{prod})$.
$Model_2 \leftarrow ML\_Model\_create(LG2, Data_{prod})$.

Generate the predicted response:
$Test_1 \leftarrow predict(Model_1, Data_{input})$
$Test_2 \leftarrow predict(Model_2, Data_{input})$
$Validation\_Response \leftarrow ExtractResponse(Data_{input})$

Return($Test_1, Test_2, Validation\_Response, \mathcal{C}$)

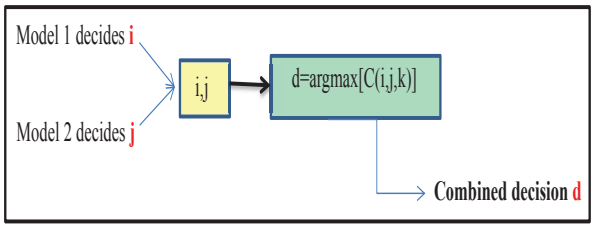

Fig. 3. Combined decision.

are used to generate the set of classes and the predicted data respectively.

2) Confusion cube construction:

To complete the *collaborative filtering*, SPC creates a **confusion cube** C which summarizes all the predictions made by the two models where:

$$C(i,j,k) = \text{Number of occurrence of the event } E_{i,j,k} \quad (1)$$

$$E_{i,j,k} = \begin{cases} \text{Model1 decides } i \\ \text{Model2 decides } j \\ \text{Correct decision is } k. \end{cases} \quad (2)$$

The number of correct and incorrect predictions are summarized with count values and broken down by each class. For each packet, we take all the possible combinations of the two models and the class having more votes will be selected. Algorithm 2 first combines all the decisions made by the two learning algorithms with the correct decision using the *combine* function. Then, it computes the number of occurrence of the event $E_{i,j,k}$ using $ComputeSum$ function. This number will be inserted in the cube for the SPC decision later (Figure 3).

Algorithm 2 Confusion cube

Input:
$Test_1$: Decision of model 1
$Test_2$: Decision of model 2.
$\mathcal{S}$: The set of the C_N possible classes.
$Validation\_Response$: the predicted data.

$D = combine(Test_1, Test_2, Validation\_Response)$
for $c_i \leftarrow 1$ to C_N **do**
  for $c_j \leftarrow 1$ to C_N **do**
    for $c_k \leftarrow 1$ to C_N **do**
      $E_{i,j,k} = Count\_Occurences\Big(\mathcal{S}(c_i), \mathcal{S}(c_j), \mathcal{S}(c_k), D\Big)$
      $C(c_i, c_j, c_k) = E_{i,j,k}$
    end for
  end for
end for

Return(C: confusion cube)

3) SPC model creation:

By considering the decisions made by the two ML techniques, we derive the confusion cube that provides the classification capability for the SPC model. The proposed model uses algorithm 3 to make a final decision d which is the decision that has gained more votes.

Algorithm 3 SPC model creation

Input :
C : confusion cube.
$\mathcal{S}$: The set of the C_N possible classes.

---

for $c_i \leftarrow 1$ to C_N **do**
  for $c_j \leftarrow 1$ to C_N **do**
    $c_k \leftarrow argmax(C(c_i, c_j), 1..C_N)$
    $SPC\_Model\Big(\mathcal{S}(c_i), \mathcal{S}(c_j)\Big) \leftarrow \mathcal{S}(c_k)$
  end for
end for

Return(SPC_Model)

Figure 4 shows an example of confusion cube creation where model 1 decides the class c_2 and model 2 decides the class c_3 ($i = c_2$ and $j = c_3$). The final decision will be the class having more vote which is class 2 c_2 (Figure 5).

D. Test model

A new data was used in comparison tests with SPC classification. The created model should give as many true positives and true negatives as possible before implementing it in the network.

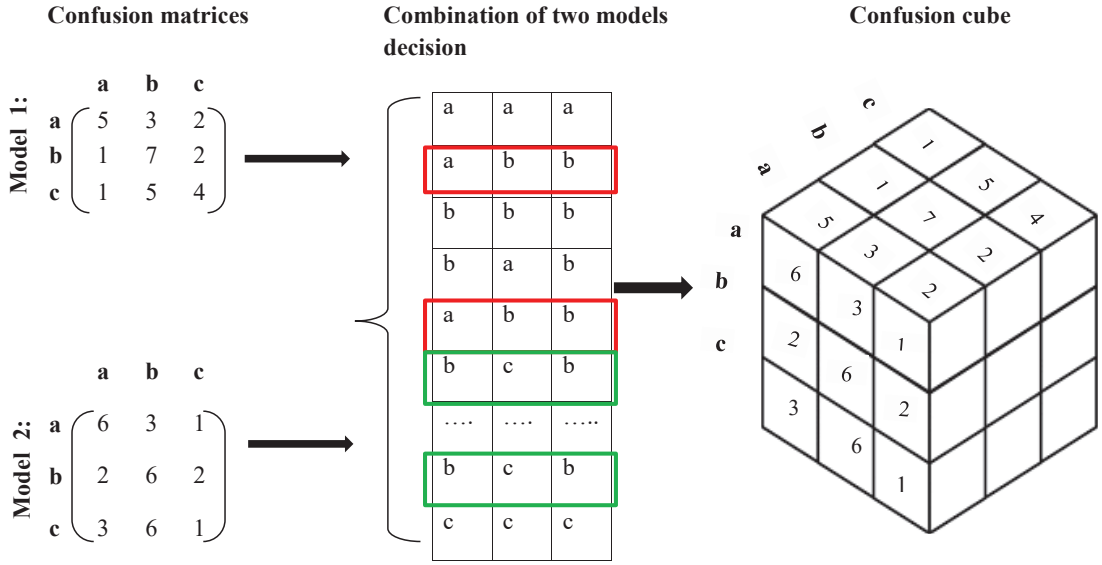

Fig. 4. Confusion cube creation.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
(i=C_2, j= C_3)	3	30	10	5	2	9	0	1	0	0

Fig. 5. An example of class selection for SPC model.

TABLE I
DATASET STATISTICS

Classes	Training	Testing
Normal	56000	37000
Analysis	2000	677
Backdoor	1743	583
DoS	12264	4089
Exploits	33393	11132
Fuzzers	18184	6062
Reconnaissance	10491	3496
Shellcode	1133	378
Worms	130	45

III. EXPERIMENTS RESULTS

A. Data description

UNSW−NB−15 contains 9 types of attacks, which are Analysis, Backdoor, DoS, Exploits, Fuzzers, Normal, Reconnaissance, Shellcode, and Worms. UNSW−NB−15 is composed of two parts: training set "UNSW−NB−15 training−set.csv" and testing set, "UNSW−NB15−testing−set.csv". The number of records in the training set is 175,341 and 82,332 in the testing set [12]. Table I shows the number of packets in the training and testing set for each class [13].

B. ML selection: best two ML algorithms for SPC model

Figure 6 shows the best combination decision architectural model. We remark that ensemble boosted trees and complex tree have a higher combined accuracy compare to the other combined techniques. Figure 6 shows the effect of α on the selected algorithms' combination performance. For a better combination, we evaluate the performance of all the algorithms. We can remark that usually, any combination with a complex tree gives better accuracy compared to the other algorithms (see Table II). Also, α has a big impact on the learning accuracy. For our work, accuracy is not only the decision criteria, the detection probability is also an important criteria. In fact, the two combined techniques should achieve a good detection rate for each attack type. After completing the off-line data processing step, we chose to combine: **Complex Tree** and **Ensemble Boosted Tree** with $\alpha = 0.5$.

TABLE II
SOME EXAMPLES FOR DIFFERENT VALUES OF α AND ML ALGORITHMS

ML technique 1	ML technique 2	α	accuracy
Complex Tree	Ensemble: Boosted Trees	0.5	80.43
Complex Tree	Quadratic SVM	0.53	80.86
Complex Tree	Ensemble: Bagged Trees	0.575	81.1

Table II presents some examples for different values of α and learning algorithms.

C. SPC performance

Figure 8 shows the TNR and FPR rate of the SPC model compared to the complex decision and ensemble boosted tree. The TNR of SPC is equal to 96% with an FPR equals to 4% meaning that SPC has a low proportion of normal packets that

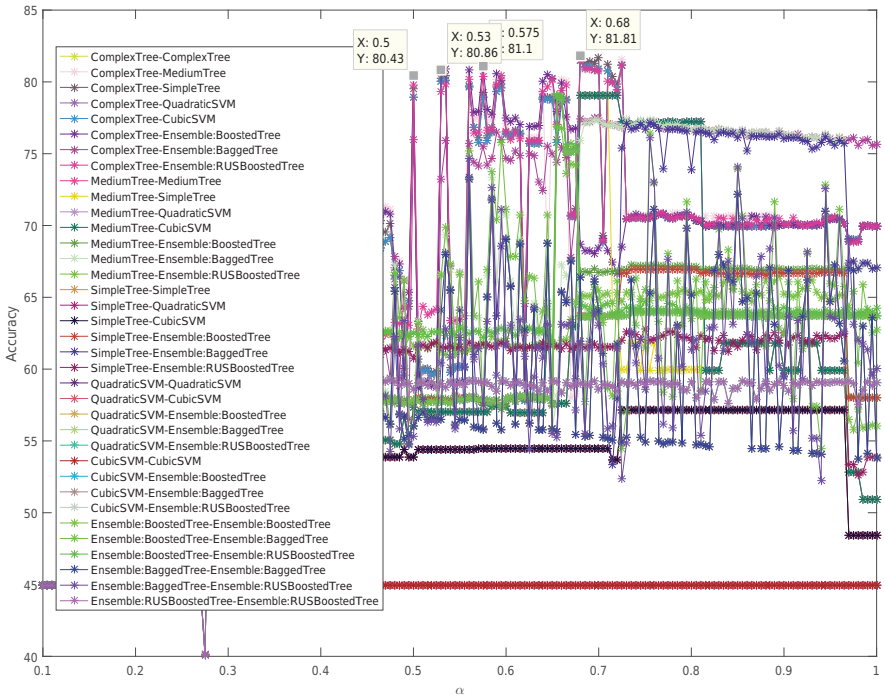

Fig. 6. Effect of α on the learning algorithms combination.

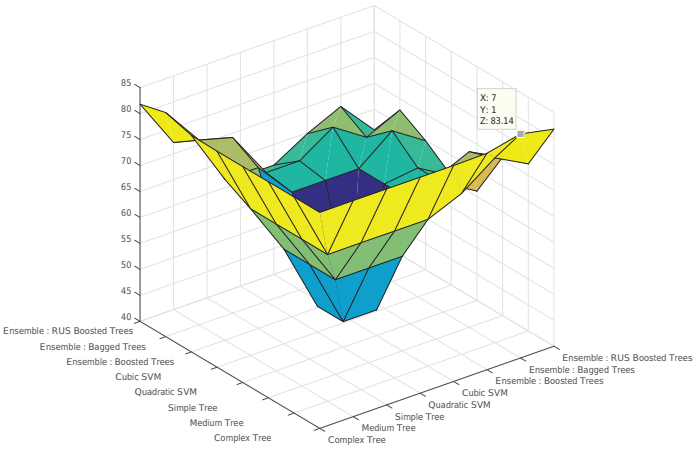

Fig. 7. Best combination decision architectural model.

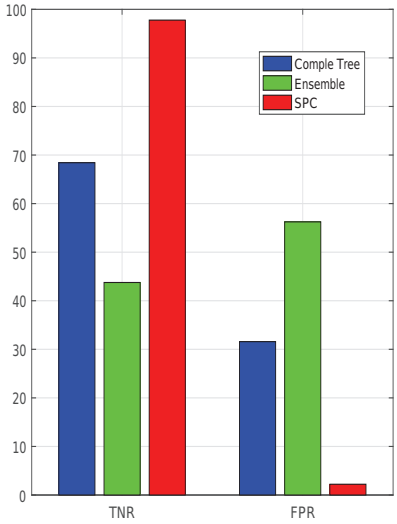

Fig. 8. TNR and FPR rate of SPC model compared to complex decision and ensemble boosted tree.

are not correctly identified. Compared to the ensemble boosted tree, the FPR is equal to 55%. So more than half of the normal packets are not detected.

Figure 9 shows the recall, precision and F1 rate for the SPC model compared to complex decision and ensemble boosted tree. The precision rate shows that SPC detects better the malicious packets compared to the complex decision and ensemble boosted tree. We can see that ensemble boosted tree has the lowest F1 function compared to the complex tree and SPC model. SPC, on the other hand, achieves 92% of F1 which reveals the good performance of SPC in terms of attack detection.

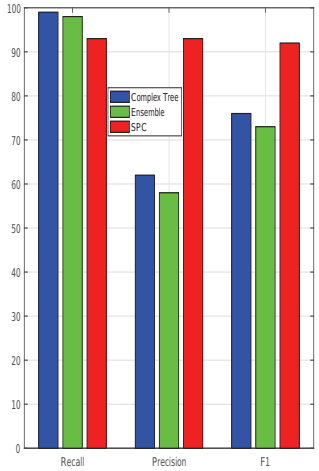

Fig. 9. Recall, precision and F1 of SPC model compared to complex decision and ensemble boosted tree.

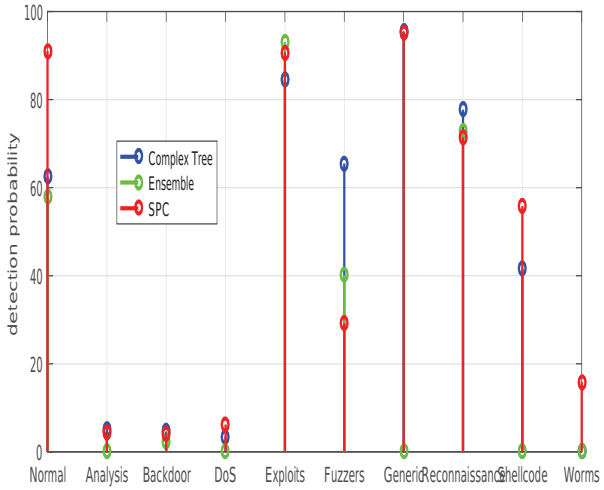

Fig. 10. the detection rate of SPC model compared to complex decision and ensemble boosted tree.

Figure 10 shows the detection rate of the SPC model compared to the complex decision and ensemble boosted tree for 10 classes classification. It can be seen that also in terms of detection rate; SPC is able to truly detect the normal packets with an accuracy of 94% however complex tree and ensemble boosted tree detect only 62% and 59% of packets respectively which represents more than 30% in normal packets detection for the SPC. We remark also that SPC achieves a good detection rate for the other different attack types.

IV. Conclusion

In this paper, we present a new model called SPC for anomaly detection in the cloud computing environment. SPC mixes the advantages of both learning schemes in machine learning methods for anomaly detection and attack classification in order to propose the required countermeasure for each attack. The UNSW dataset has been used to evaluate the SPC performance. Results demonstrate that the proposed model can achieve 81% accuracy with anomaly detection meaning that 20% better than the traditional approaches.

Acknowledgment

This work was supported by Qatar University Internal Grant IRCC-2020-001. The statements made herein are solely the responsibility of the author[s].

References

[1] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, and Y. Xiang, "Block design-based key agreement for group data sharing in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2018.

[2] Zina Chkirbene, Sebti Foufou, Ridha Hamila, Zahir Tari, and Albert Y. Zomaya, "Lacoda: Layered connected topology for massive data centers," *Journal of Network and Computer Applications*, vol. 83, pp. 169 – 180, 2017.

[3] Lav Gupta, Raj Jain, Mohammed Samaka, Aiman Erbad, and Deval Bhamare, "Performance evaluation of multi-cloud management and control systems," *Recent Advances in Communications and Networking Technology (Formerly Recent Patents on Telecommunication)*, vol. 5, no. 1, pp. 9–18, 2016.

[4] Muhammad Abedin, Syeda Nessa, Latifur Khan, and Bhavani Thuraisingham, "Detection and resolution of anomalies in firewall policy rules," in *Data and Applications Security XX*, Ernesto Damiani and Peng Liu, Eds., Berlin, Heidelberg, 2006, pp. 15–29, Springer Berlin Heidelberg.

[5] Aiman Erbad, Norman C Hutchinson, and Charles Krasic, "Doha: scalable real-time web applications through adaptive concurrent execution," in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 161–170.

[6] Giovanni Vigna, William Robertson, and Davide Balzarotti, "Testing network-based intrusion detection signatures using mutant exploits," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, New York, NY, USA, 2004, CCS '04, pp. 21–30, ACM.

[7] Zina Chkirbene, Aiman Erbad, Ridha Hamila, Amr Mohamed, Mohsen Guizani, and Mounir Hamdi, "Tidcs: A dynamic intrusion detection and classification system based feature selection," *IEEE Access*, vol. 8, pp. 95864–95877, 2020.

[8] Deval Bhamare, Maede Zolanvari, Aiman Erbad, Raj Jain, Khaled Khan, and Nader Meskin, "Cybersecurity for industrial control systems: A survey," *computers & security*, vol. 89, pp. 101677, 2020.

[9] Zina Chkirbene, Aiman Erbad, Ridha Hamila, Ala Gouissem, Amr Mohamed, Mohsen Guizani, and Mounir Hamdi, "Weighted trustworthiness for ml based attacks classification," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–7.

[10] Aliya Tabassum, Aiman Erbad, and Mohsen Guizani, "A survey on recent approaches in intrusion detection system in iots," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 1190–1197.

[11] Zina Chkirbene, Aiman Erbad, Ridha Hamila, Ala Gouissem, Amr Mohamed, and Mounir Hamdi, "Machine learning based cloud computing anomalies detection," *IEEE Network*, vol. 34, no. 6, pp. 178–183, 2020.

[12] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov 2015, pp. 1–6.

[13] Zina Chkirbene, Sohaila Eltanbouly, May Bashendy, Noora AlNaimi, and Aiman Erbad, "Hybrid machine learning for network anomaly intrusion detection," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. IEEE, 2020, pp. 163–170.