Adaptive User Profiling with Online Incremental Machine Learning for Security Information and Event Management

Dilli P. Sharma *, Barjinder Kaur *, Farzaneh Shoeleh *, Masoud Erfani*, Duc-Phong Le †,
Arash Habibi Lashkari *, Ali A. Ghorbani *
*Canadian Institute for Cybersecurity, University of New Brunswick, NB, Canada
E-mail: {dilli.sharma, kaur.barjinder, farzaneh.shoeleh, masoud.erfani, a.habibi.l, ghorbani}@unb.ca
† Bank of Canada, Ottawa, Canada, E-mail: dle@bankofcanada.ca

***Abstract*—In the past few years, there has been an exponential growth in network and Internet traffic. This trend will continue to increase due to digitalization and resulting in more interconnectivity among the users. Due to this, more data has started being treated as streaming data. This data distribution, mostly non-stationary, high-speed, and infinite length, contains information regarding user activities. Thus, it is essential to provide an anomaly detection model that can deal with the evolving nature of data, update, adapt, and give system administrators timely action and minimize false alarms. This paper proposes a dynamic and adaptable user profiling for security information and event management system using online incremental machine learning. An anomaly detection-based user profiling technique dynamically learns users' activities and updates their profiles over time. The experiments to detect anomalous activities is performed on datasets generated in realistic scenario based on user's activities and recorded in three different time windows (e.g., 30-minutes, 1-hour, and 2-hour) of a month. The system's efficacy is evaluated with the Isolation Forest *(iForest)* approach to detect anomalies in incremental learning settings for all the datasets. We further compared the performance of our proposed incremental approach with a non-incremental baseline model in terms of the detection of abnormal user activities. The experimental results show that our proposed incremental model outperformed its baseline counterpart model. It can be used more opportunistically to profile users as a component of Security Information and Event Management (SIEM) systems.**

***Index Terms*—Machine learning; anomaly detection; cybersecurity; user profiling; incremental learning**

I. Introduction

Internet is said to be the core pillar where all the information can be easily and readily available. With the advancement in Internet technologies, more users are getting themselves connected to this technology. The recent study came in January 2021, highlighting the stats that there were 4.66 billion active Internet users worldwide [6]. So, there is an unprecedented amount of data presented from different domains which help users in one way or another. But this has led to an increase in cybercrime either network intrusion or posing a threat by performing different malicious activities from both inside and outside of the organization.

This shows that although solutions are being provided for securing the data, the organizations lack capturing the user experience. A special report published in 2020 measured cybercrime costs to grow by 15 percent per year over the next five years, reaching $10.5 trillion USD annually by 2025, up from $3 trillion USD in 2015 [11]. Due to the easy accessibility of the devices and connectivity over the network, different kinds of applications are run on the machine; the same machine could also be used to browse different websites. Simultaneously, logs are generated that capture profile of a user. Thus, constructing a user profile is one such important concept that has become the need of the hour and needs to be built dynamically using users' activities. This profile based on users' activities collected from different sources will further help organizations to detect anomalous activity, generate alerts, and change policies according to it. An approach proposed by Lashkari et al. [8] creates a new user profile from all the available sources. After gathering users' information based on different profiling criteria, the authors created a security profile of a user. Similarly, a recommendation system is proposed for Google News, where each user's profile is updated and built based on their click history [3]. However, user behavior is unpredictable, i.e., the system needs to be monitored continuously. Also, it has become essential to design a system that can detect significant deviations in data and provide user-oriented service in real-time.

In this work, we propose an anomaly detection-based user profiling that dynamically learns from the user activities and updates the model. Fig. 1 depicts steps of our proposed framework, which we followed in this study to develop this adaptive user profiling model. The steps are defined as follows:

1) The data source is prepared, which included data recorded from three different user activities, i.e., web-browsing, network, and process-based activities.
2) Based on the activities data recorded, three different datasets such as 1Month_30minutes, 1Month_1H, 1Month_2H were prepared, and all the datasets have all the records of three activities.
3) The raw data is further normalized, i.e., preprocessed.
4) Further, features are extracted from three different categories, which are divided into general, network, and

 ISBN: 978-1-61208-919-5

process (application) based; these features are explained in Section V.

5) The selected features were fed into the machine learning approach to perform experiments.
6) Finally, the results are comparatively analyzed and presented with a $'non-incremental'$ approach and our proposed online $'incremental'$ approach.

This proposed adaptive user profiling system updates the data according to the dynamic changing behavior of the user. The **key contributions** of this work are as follows:

- We propose a dynamic and adaptable user profiling with online incremental machine learning for the Security Information and Event Management (SIEM) system.
- Secondly, we have analyzed the results on three different datasets (e.g., 1Month_30min, 1Month_1H, 1Month_2H)
- Finally, we compared the performance of our proposed approach with a baseline non-incremental model.

The rest of the paper is organized as follows: Section II briefly recalls the related works for user profiling. Section III discusses data preprocessing and anomaly detection classifier used in this work. Section IV presents our proposed incremental approach. In Section V, the dataset details, experimental setup and results analysis are presented. Finally, the conclusion and future directions are discussed in Section VI.

II. Related Work

In the recent past, research has been bent towards analyzing user behavior and building profiles in real-time [23][24]. However, most of the existing studies report their results using either static dataset or did not incrementally update and adapt, i.e., baseline model according to the changes noticed.

In [19] authors presented an adaptive search system based on user profile. The information is collected from browsing history for constructing the user's profile, and the update is performed whenever a change is noticed in browsed web pages. The search results should be adapted to users with different information need.

An insider-threat detection model based on user behavior has been proposed in [7]. The behavior is analyzed from the collected dataset, i.e., user's daily activity summary, e-mail contents topic distribution, and user's weekly e-mail communication history. The abnormal behavior is detected using four different one-class machine learning algorithms Gaussian density estimation (Gauss), Parzen window density estimation (Parzen), Principal Component Analysis (PCA), and K-means clustering. On the same theme of inside-threat detection, the user-profile approach has been utilized to detect anomalous behavior. Singh et al. [17] used an ensemble hybrid machine learning approach using Multi-State Long Short-Term Memory (MSLSTM) and Convolution Neural Networks (CNN) approaches to detect outlier activities from the patterns extracted from spatial-temporal behavior features.

Another study presented an unsupervised user behavior modeling based on session activities. The authors analyzed the activities using LSTM based autoencoder following a two-step process. First, it calculates the reconstruction error using the autoencoder on the non-anomalous dataset, and then it is used to define the threshold to separate the outliers from the normal data points. The identified outliers are then classified as anomalies. The CERT dataset, which is recorded using users' day-to-day activities, includes all the files about system usage, logged time, and date that has been used for research work [14]. A study has been proposed for enterprise organizations with the same dataset where user profiling is built by analyzing log authorization. To evaluate their method, the authors used Random Forest and achieved an accuracy of 97.81% for detecting the anomalous behavior of the user [25].

The authors in [22] proposed a novel Ouda's authentication framework for security purposes. The framework is built by using a user profile that captures the anomalous actions from users' activities. The information representing user activities is collected using their unique identification. Important features are selected that represent anomalous action, preprocessing performed, and results are predicted on a binary-basis, which acts as a base for building user profile. The anomaly detection technique used was implemented based on machine learning clustering algorithms [21] [20].

The anomalous behavior of the user was also detected based on similarity clustering. The authors proposed a model consisting of four components: datalog collector, data log analyzer, profile storage, and behavior detector, which performs different functionality [4]. The User and Entity Behavior Analytics (UEBA) module presented by Madhu Shashanka et al. [15] uses the Singular Values Decomposition (SVD) algorithm to detect anomalous behavior. The module built tracks and simultaneously monitors users' IP addresses and devices. The system was proposed for an enterprise network.

A knowledge-driven user pattern discovery approach was proposed to analyze user behavior in [10]. The authors extracted the patterns using audit logs from distributed medical imaging systems. These patterns help the administrators to identify the users with anomalous behavior, which may threaten the data privacy and system's integrity.

A scalable system for high-throughput real-time analysis of heterogeneous data streams was proposed in [2]. The architecture named RADISH enables the incremental development of models for predictive analytics and anomaly detection as data arrives into the system. The architecture also allows for ingesting and analysis of data on the fly, thereby detecting and responding to anomalous behavior in near real-time.

Shaman et al. [13] proposed a supervised learning-based user profiling approach using Gradient boosting. This work provides a mechanism for user identification and behavior profiling by analyzing the individual uses of each application. The application-level flow sessions were identified based on DNS filtering criteria and timing. However, the scope of this work is limited to the application level.

Most of the existing user profiling work mainly focuses on the static aspect of user behavior analysis. However, user behavior changes dynamically over time. So, the static nature of the model cannot learn and adapt to the changing behavior of the users. In this work, we devise dynamic and adaptable

 ISBN: 978-1-61208-919-5

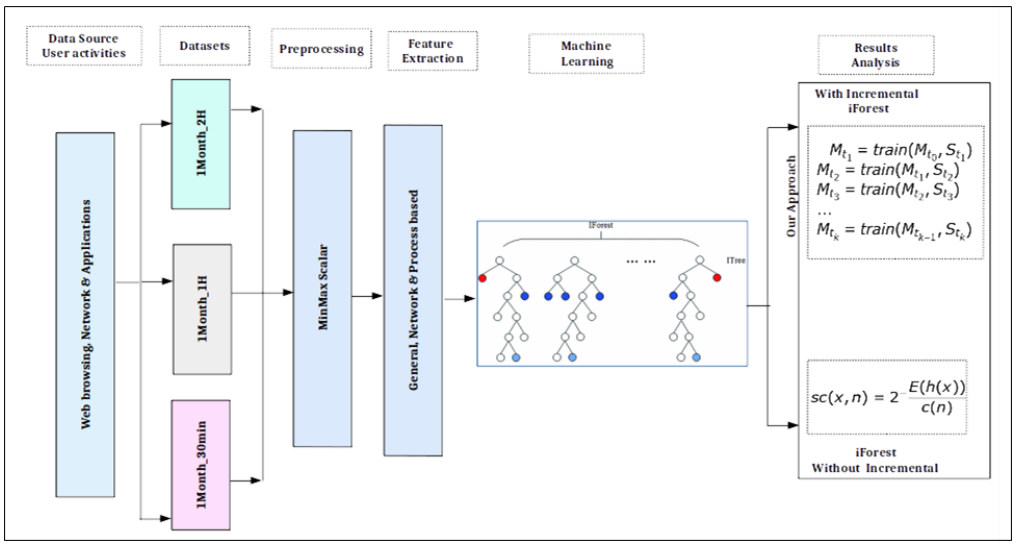


Fig. 1: User profiling framework with the proposed incremental approach.

user profiling using increment learning.

III. Preliminaries

In this section, we describe data preprocessing and Isolation Forest classifier for detecting abnormal user behaviors.

A. Data Preprocessing

As datasets contain numerical and nominal values, preprocessing the training and testing dataset is an important step. The goal is to normalize the feature values to the same scale. Our approach considers all the features of the dataset as each feature is equally important. For this we applied the MinMaxScaler object from the sklearn library [18] to rescale the values into the range*[0,1]* [1]. The MinMaxScaler can be defined using (1).

$$Xnr = \frac{F - Min_F}{Max_F - Min_F} \tag{1}$$

where Max_F and Min_F are maximum and minimum values obtained for F, which is a feature vector of the features. We replaced the null values with zero. The network data suffer from missing or null values that could also appear as outliers or wrong data, so we have replaced these null values with zero before performing our analysis.

B. Anomaly Detection using Isolation Forest Classifier

In this work, we have used Isolation Forest (iForest) [9] to detect abnormal user behaviors. It is a popular tree-based, unsupervised outlier detection approach that works on isolating outliers, i.e., anomalies. The quantitative property of iForest is they are fewer and very different from the usual instances.

It works by building an ensemble of *iTrees* from a given dataset. The algorithm takes n random samples of size from a given dataset. For each random sample, the *"iTree"* is built by splitting the sub-sample instances over a split value of a randomly selected feature so that the instances whose corresponding feature value is smaller than the split value go left. The others go right, and the process continues recursively until the tree is entirely built. The split value is selected at random between the minimum and maximum values of the selected feature. This results in a shorter tree path for outliers and is thus easy for detecting [9][12]. The anomalous score sc is calculated as defined in (2).

$$sc(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{2}$$

where $h(x)$ is the path length of sample x, and $E(h(x))$ represents the average value of $h(x)$ from *iTrees* collection. The value of $c(n)$ is the average path length of unsuccessful search in a Binary Search Tree with n nodes [9]. Then the instance x is assigned to outlier if the value of sc is close to 1 otherwise considered as normal.

IV. Proposed Online Incremental Learning Model

In this section, we present our proposed online incremental learning model. We design an incremental model where a machine evolves incrementally learning from the previously trained model with a new data block. The learning process starts from an initial baseline model. Evolving of a machine using this approach is shown in Fig. 2. Let M_{t_0} denotes an initial baseline model at a time point t_0. A sequence of model $\{M_{t_1}, M_{t_2}, M_{t_3}, \ldots, M_{t_k}\}$ is generated from the baseline M_{t_0} with incremental training on stream of data blocks $\{S_{t_1}, S_{t_2}, S_{t_3}, \ldots, S_{t_k}\}$ at time t_1, t_2 ,t_3, and t_k, respectively. The model M_{t_k} is evolving from the baseline model M_{t_0} with k total incremental updates using the data stream. This incremental model evolution is derived as follows:

$$\begin{aligned} M_{t_1} &= train(M_{t_0}, S_{t_1}) \\ M_{t_2} &= train(M_{t_1}, S_{t_2}) \\ M_{t_3} &= train(M_{t_2}, S_{t_3}) \\ &\ldots \\ M_{t_k} &= train(M_{t_{k-1}}, S_{t_k}) \end{aligned} \tag{3}$$

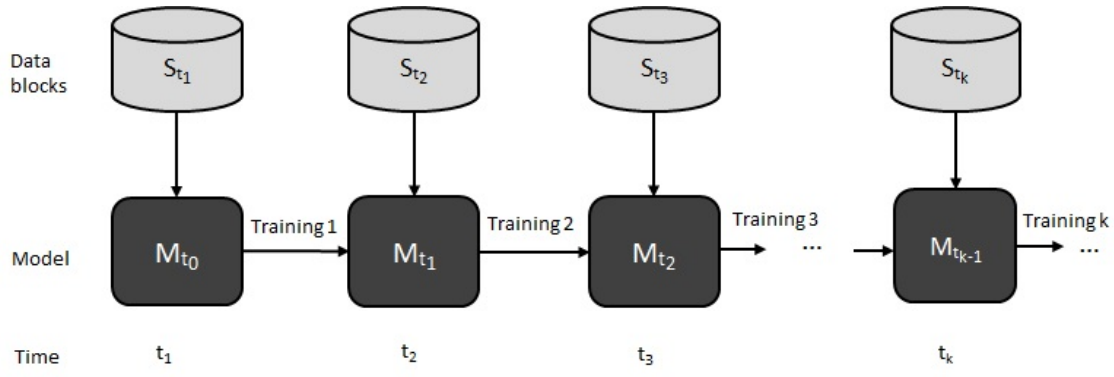


Fig. 2: Evolving a model (machine) with online incremental learning

where, $w = t_k - t_{k-1}$ is a model (re)training time window. This time window can be a fixed constant or variable time period (e.g., 1 day, 3 days, 7 days, etc.) or dynamic (event-driven). The dynamic (re)training time window is an event driven where receiving alerts from security event detectors or predictive analytic determines when the model is to be retrained.

V. Experimental Setup & Results Analysis

In this section, we present the description of dataset, experimental setup and results analysis.

A. Dataset Description

We used the dataset collected from four users performing three different activities. The activities include web browsing, network, and application. The users' activities captured has three common properties (*IP, MAC-Address, Activities*) where *IP* represents the users' IP address, and *MAC-Address* represent the machine address using which the user is performing the different activity. On the other hand, *Activities* properties include generating a users' activities: web-browsing, file transferring, and opening an application.

The users' activities have been recorded between 8:00 AM to 5:00 PM. Furthermore, the occurrences of the users' activities have been randomly considered, while it would be possible that the user performed two kinds of activities together. The number of activities for each user has been identified based on predefined averages and standard deviations. Regarding being a normal or abnormal activity, various types have been defined for users' activities.

After generating user scenarios that include their activities, the generated file has been processed by another implemented module that produces the final STIX format. The module reads, analyzes, and identifies the type of activities in the JSON file. Table I presents the details of simulated datasets. The features are extracted from STIX format file which are categorised into *general*, *network-based*, *process-based*. Here, the network features include all those features related to network activities of the user, process-based features include the information regarding file transfer, process values, whereas the general feature summarizes the minimum and maximum

TABLE I: SUMMARY OF THE DATASETS.

Name	Time period	Session duration	Number of instances	Features
1Month_30min	One month	30 minutes	2280	44
1Month_1H	One month	1 hour	1116	44
1Month_2H	One month	2 hours	560	44

session of a user. More detailed description of the dataset can be found in [16].

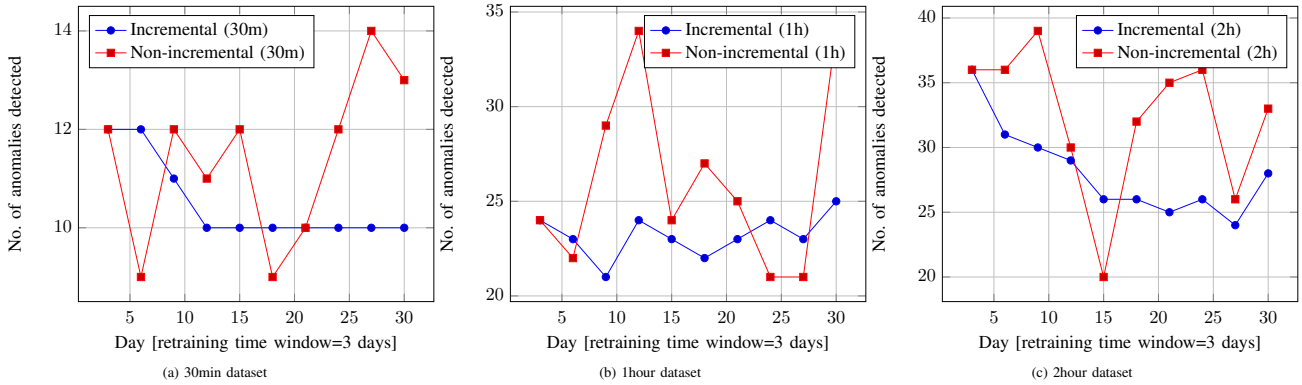
B. Experimental Setup

We used a standard sklearn [18] Application Programming Interface (API) for the experimentation. As described in Section III-B, we trained Isolation Forest (iForest) classifier for the anomaly detection with the three-datasets as presented in Table I. Each dataset is randomly divided into training and testing sets with 90% (training) and 10% (testing). Each training data is further divided into ten blocks, each with a time window of three days. Then, we trained both our incremental iForest model and a baseline iForest model with each data block. The baseline model is retrained with each new data block only, whereas; the proposed model is trained and updated incrementally with each new data block adding more estimators. Below is the description of model fitting and updating incrementally over time using the proposed approach.

Incremental Training & Updating: During training, we incrementally trained our model with the new data samples over time. Initially, a model is created using the iForest classifier with some initial number of base estimators in the ensemble ($n\_estimators = 100$) and $warm\_start = True$. A setting of $warm\_start$ to $True$ enables us to continue training to the previously trained model and add more estimators to the ensemble. Before every training, we increment the $n\_estimators$ parameter and update it to the model using the $set\_params()$ method. The trained model is persistently saved/loaded to/from a file on disk. We use joblib [5] API for reading or reconstructing a Python object of the model from a file persisted dump.

The performance of each evolving model (machine) is evaluated with the same test data after each incremental/(re)training. We stored the anomalies detected after each re/training of our proposed and a baseline model. An increase or decrease in the number of anomalies in each iterative (re)training step can help us measure the model's performance.

 ISBN: 978-1-61208-919-5



(a) 30min dataset (b) 1hour dataset (c) 2hour dataset

Fig. 3: Comparison of results of our proposed incremental model with a non-incremental model.

The obtained results are comparatively analyzed and discussed in Section V-C.

C. Results Analysis

Here, we present the results analyzed using our proposed incremental approach where the system dynamically learns and adapts according to changes noticed in user activities.
A comparison of results in terms of several anomalous activities detected with the proposed incremental training model and a baseline non-incremental model using three different datasets can be seen in Fig. 3, where the x-axis represented the day of a month when we trained the model and the y-axis represents the number of anomalies detected.

In Fig. 3 (a) we present the results of 1Month_30min dataset. A comparative analysis performed between proposed incremental approach with a baseline non-incremental using iForest model as discussed in Section IV shows that several detected anomalies significantly decreases with increasing the number of incremental training and it converges to a steady after fourth incremental training (i.e., after 12 days) using the proposed approach. However, in the non-incremental settings, the number of anomalies detected was found to be inconsistent and unpredictable.

Further, we analyzed and compared the results with two datasets, i.e., 1Month_1H and 1Month_2H. The results presented in Fig. 3 (b) and (c) shows that although the detected number of anomalies slightly decreases using the proposed model, there are some fluctuations since the incremental model requires more iterations of training and more data to learn and adapt with user activities data collected in a longer period of time (i.e., large time window). Also, there are larger changes and fluctuations in Fig. 3 (b) and (c) since training with the new data only poses a concept drift. The results show that the small-time window provides more insight into user activities as large-time windows are more diverse to changes.

An in-depth analysis has been performed using the proposed online incremental approach in terms of whether all the anomalies detected by the i+1th model are from the anomalies detected in the previous ith model or new anomalies. For this, we analyzed the results obtained after conducting experiments on all three datasets and presented them as a summary in Table II. Each row of the table has three components, first, no. of anomaly detected, second no. of new anomalies anomaly detected, and an arrowhead/hyphen. For example, the first row of the 3rd iteration column has $11(0)\downarrow$ where 11 is a no. of anomaly detected in a 3rd machine, (0) indicates no new anomaly detected in this machine that is all the detected 11 anomalies are from the previous machine (2nd machine), and $\downarrow$ represents detected no. of anomalies are less (decreased) than the previous model. Similarly, $\uparrow$ represents no. of anomalies increases, - (hyphen) means no change. Color 'green' shows a machine is performing well as expected, 'blue' color indicates a machine is good but not desired because the machine detected lower no. of anomalies, but it also detected new anomalies. 'Red' color represents a machine performing not well. Results show that our incremental model perfectly leaned as expected with 1Month_30min datasets since there are neither more anomalies detected nor new anomalies in each consecutive training. However, the non-incremental model has many fluctuations. With all three-datasets, our proposed incremental iForest model shows significantly good performance as compared to the baseline non-incremental iForest model.

VI. Conclusion

The biggest challenge nowadays is to identify the malicious activities which are increasing due to inter-connectivity among users and the devices. Also, it has been noticed due to this global pandemic, a greater number of users are accessing office networks for communication, transferring files sitting back at home. This has resulted in an upsurge in hidden attacks as malicious users are trying to access the system in one way or another. In this dynamic changing environment where everything is non-stationery and data distribution is changing faster, building a user profile helps in identifying the intentions of these users and taking timely action to prevent further harm. To overcome this issue, the following are the key findings obtained from this study which can help us build a user profiling system that adapts and raise an alert when there is a slight deviation in the system:

- We proposed an online incremental anomaly detection-based user profiling model for SIEM systems. The pro-

TABLE II: SUMMARY OF THE COMPARATIVE PERFORMANCE ANALYSIS OF RESULTS.

Model	Datasets	# of anomalies, # of new anomalies after each (re)training									
		1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Our Incremental using iForest Model	1month_30min	12	12 (0) -	11 (0) ↓	10 (0) ↓	10 (0) -	10 (0) -	10 (0) -	10 (0) -	10 (0) -	10 (0) -
	1month_1h	24	23 (2) ↓	21 (0) ↓	24 (4) ↑	23 (0) ↓	22 (0) ↓	23 (1) ↑	24 (2) ↑	23 (0) ↓	25 (2) ↑
	1month_2h	36	31 (1) ↓	30 (1) ↓	29 (1) ↓	26 (0) ↓	26 (0) -	25 (0) ↓	26 (1) ↑	24 (0) ↓	28 (4) ↑
Non-Incremental using iForest Model	1month_30min	12	9 (0) ↓	12 (3) ↑	11 (0) ↓	12 (2) ↑	9 (0) ↓	10 (1) ↑	12 (2) ↑	14 (2) ↑	13 (0) ↓
	1month_1h	24	22 (2) ↓	29 (11) ↑	34 (12) ↑	24 (1) ↓	27 (7) ↑	25 (3) ↓	21 (2) ↓	21 (6) -	34 (16) ↑
	1month_2h	36	26 (1) ↓	39 (16) ↑	30 (3) ↓	20 (1) ↓	32 (17) ↑	35 (12) ↑	36 (7) ↑	26 (2) ↓	33 (9) ↑

posed model dynamically learns from the user activities and updates the model incrementally over time.

- We validated the performance of the proposed incremental approach against the non-incremental model in terms of adaptability of user activities for 3-different datasets.
- The experimental results proved that our proposed incremental model outperformed its counterpart model.
- Our findings suggest that the proposed model should be applied more opportunistically to profile users as a SIEM system component.

ACKNOWLEDGEMENT

The authors graciously acknowledge the funding support from the NSERC CRD Grant, Tier 1 Canada Research Chair, and IBM to Dr. Ghorbani.

REFERENCES

[1] I. Apostol, M. Preda, C. Nila, and I. Bica, "IoT Botnet Anomaly Detection Using Unsupervised Deep Learning," *Electronics*, vol. 10, no. 16, p. 1876, 2021.

[2] B. Böse, B. Avasarala, S. Tirthapura, Y.-Y. Chung, and D. Steiner, "Detecting insider threats using radish: A system for real-time anomaly detection in heterogeneous data streams," *IEEE Systems Journal*, vol. 11, no. 2, pp. 471–482, 2017.

[3] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: scalable online collaborative filtering," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 271–280.

[4] S. Hu, Z. Xiao, Q. Rao, and R. Liao, "An anomaly detection model of user behavior based on similarity clustering," in *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE, 2018, pp. 835–838.

[5] Joblib, "Joblib: Running Python Functions as Pipeline Jobs," 2021, https://joblib.readthedocs.io/en/latest/, Accessed on 2021-09-10.

[6] J. Johnson, "digital- population-worldwide," 2021, urlhttps://www.statista.com/statistics/617136/digital-population-worldwide/, Accessed on 2018-09-20.

[7] J. Kim, M. Park, H. Kim, S. Cho, and P. Kang, "Insider threat detection based on user behavior modeling and anomaly detection algorithms," *Applied Sciences*, vol. 9, no. 19, p. 4018, 2019.

[8] A. H. Lashkari, M. Chen, and A. A. Ghorbani, "A survey on user profiling model for anomaly detection in cyberspace," *Journal of Cyber Security and Mobility*, pp. 75–112, 2019.

[9] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[10] W. Ma, K. Sartipi, and D. Bender, "Knowledge-driven user behavior pattern discovery for system security enhancement," *International Journal of Software Engineering and Knowledge Engineering*, vol. 26, no. 03, pp. 379–404, 2016.

[11] S. Morgan, "Cybercrime to cost the world 10.5 trillion annually by 2025," 2021, https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/, Accessed on 2018-09-21.

[12] R. C. Ripan, I. H. Sarker, M. M. Anwar, M. Furhad, F. Rahat, M. M. Hoque, M. Sarfraz *et al.*, "An isolation forest learning based outlier detection approach for effectively classifying cyber anomalies," in *International Conference on Hybrid Intelligent Systems*. Springer, 2020, pp. 270–279.

[13] F. Shaman, B. Ghita, N. Clarke, and A. Alruban, "User profiling based on application-level using network metadata," in *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, 2019, pp. 1–8.

[14] B. Sharma, P. Pokharel, and B. Joshi, "User Behavior Analytics for Anomaly Detection Using LSTM Autoencoder-Insider Threat Detection," in *Proceedings of the 11th International Conference on Advances in Information Technology*, 2020, pp. 1–9.

[15] M. Shashanka, M.-Y. Shen, and J. Wang, "User and entity behavior analytics for enterprise security," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1867–1874.

[16] F. Shoeleh, M. Erfani, S. S. Hasanabadi, D.-P. Le, A. H. Lashkari, and A. Ghorbani, "User Profiling on Universal Data Insights tool on IBM Cloud Pak for Security," in *Proceedings of the 18th International Conference on Privacy, Security and Trust (PST2021)*, 2021.

[17] M. Singh, B. M. Mehtre, and S. Sangeetha, "User behavior profiling using ensemble approach for insider threat detection," in *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*. IEEE, 2019, pp. 1–8.

[18] Sklearn, "Scikit-Learn: Machine Learning in Python," 2021, https://scikit-learn.org/stable/, Accessed on 2021-08-18.

[19] K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive web search based on user profile constructed without any effort from users," in *Proceedings of the 13th international conference on World Wide Web*, 2004, pp. 675–684.

[20] I. I. A. Sulayman and A. Ouda, "Data analytics methods for anomaly detection: Evolution and recommendations," in *2018 International Conference on Signal Processing and Information Security (ICSPIS)*. IEEE, 2018, pp. 1–4.

[21] ——, "User modeling via anomaly detection techniques for user authentication," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2019, pp. 0169–0176.

[22] ——, "Designing security user profiles via anomaly detection for user authentication," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2020, pp. 1–6.

[23] B. Veloso, B. Malheiro, J. C. Burguillo, J. Foss, and J. Gama, "Personalised dynamic viewer profiling for streamed data," in *World Conference on Information Systems and Technologies*. Springer, 2018, pp. 501–510.

[24] J. Yu, F. Liu, and H. Zhao, "Building user profile based on concept and relation for web personalized services," in *International Conference on Innovation and Information Management*. Citeseer, 2012.

[25] Z. Zamanian, A. Feizollah, N. B. Anuar, L. B. M. Kiah, K. Srikanth, and S. Kumar, "User profiling in anomaly detection of authorization logs," in *Computational Science and Technology*. Springer, 2019, pp. 59–65.

 ISBN: 978-1-61208-919-5