International Journal of Computing and Artificial Intelligence

E-ISSN: 2707-658X
P-ISSN: 2707-6571
www.computersciencejournals.com/ijcai
IJCAI 2024; 5(1): 42-49
Received: 13-12-2023
Accepted: 21-01-2024

Hanan Ismael Tarab
Department of Computer Engineering, Information Technology Collage, Altinbas University, Turkey

Corresponding Author:
Hanan Ismael Tarab
Department of Computer Engineering, Information Technology Collage, Altinbas University, Turkey

Cyber-attack detection and identification using deep learning

Hanan Ismael Tarab

DOI: https://doi.org/10.33545/27076571.2024.v5.i1a.82

Abstract

The quantity and complexity of cyber-attacks are both on the rise. For defenses to keep up with the ever-evolving threats, it will need ever-greater technological advances and fresh ideas. Traditional security methods like intrusion detection and deep packet inspection are still used and recommended, but they are not enough to keep up with the growing number of security threats. The widespread usage of APT's communication networks makes them more susceptible to assaults by cybercriminals, with possibly catastrophic outcomes. Critical infrastructures are monitored and controlled by Advanced Persistent Attack using real-time monitoring to identify aberrant behaviors of the system. The most existing Advanced Persistent Threat defenses were created to protect IT infrastructure and are seldom useful in more robust settings like factories. The primary objective of this thesis is to use various learning-based approaches to evaluate network traffic and sensory measures in real-time in order to identify and locate cyber-attacks. In order to achieve this, numerous learning-based models are presented, such as a self-tuning and scalable deep learning and classification model for cyber-attack site identification and an ensemble deep learning-based cyber-attack detection method for unbalanced Advanced Persistent Threat datasets. Two real-world advanced persistent threat datasets are used to assess the effectiveness of the suggested models. In terms of f1-score, recall, and accuracy, the models presented do better than the most recent research.

Keywords: Machine learning, cyber-attack, network security, python, advanced persistent attack, data preprocessing, and classification

1. Introduction

The act of trying to determine who was responsible for a piece of malicious software or a piece of code that was used in a cyber-attack is referred to as "cyber-attack attribution." The process of attributing assaults in cyberspace has grown more important as the number of cyber-attacks has increased. Reverse engineering is one method that may be used to determine who is responsible for a cyber-assault. We are able to obtain data about the malware executable file, such as the date it was created, the variable names that were utilized, and the library calls that were imported, from the metadata that accompanies the file. In an attribution analysis, these pieces of information might serve as features. In order to attribute the assaults, we will need to first extract the elements from the malware that may be utilized for attribution and then analyze those features using some method.

There are more and more security problems involving Industrial Control Systems (ICS) as digitalization advances at an accelerated pace without regard to security [1, 2]. Since its inception, the ICS has been protected from cyber-attacks by adopting proprietary software structures and communication protocols. By using common communication interfaces and standard software installed in distant instruments, ICS manufacturers are progressively introducing off-duty management and almost autonomous operations. It is now more susceptible to cyber-attacks from both within and outside of ICS facilities. There are a number of essential infrastructure facilities that manage smaller power processes such as railway stations, airports, and manufacturing plants [4].

[5, 6] In today's world, critical infrastructure relies on Internet of Things (IoT)-enabled cyber-physical systems to connect and interact with various items and systems all over the world [7, 8]. Although IoT technologies have many advantages for ICS, hackers and cybercriminals also have a lot to gain from them [9]. People's health and safety, industrial processes, and financial resources can all be jeopardized if a cyber-attack on vital infrastructure is successful [3]. As depicted in Fig 1, critical infrastructure facilities have been subjected to

cyber-attacks since 2010 [10, 11]. Stuxnet was one of the most well-known cyber-attacks, targeting Iranian nuclear enrichment centrifuges in 2010 [10, 11]. Zero-day exploits were mounted on a USB drive and malicious code was introduced into Siemens PLCs, causing centrifuges to spin far faster than planned [10]. In the interim, the malware alters sensor readings to hide the attack of the operators themselves. Early cyber-attacks on critical infrastructure installations have uncovered ICS security flaws and the threats they entail. During the Black-Energy attack in 2015, Ukraine's power grid was targeted, triggering a significant power outage that affected 230,000 people.

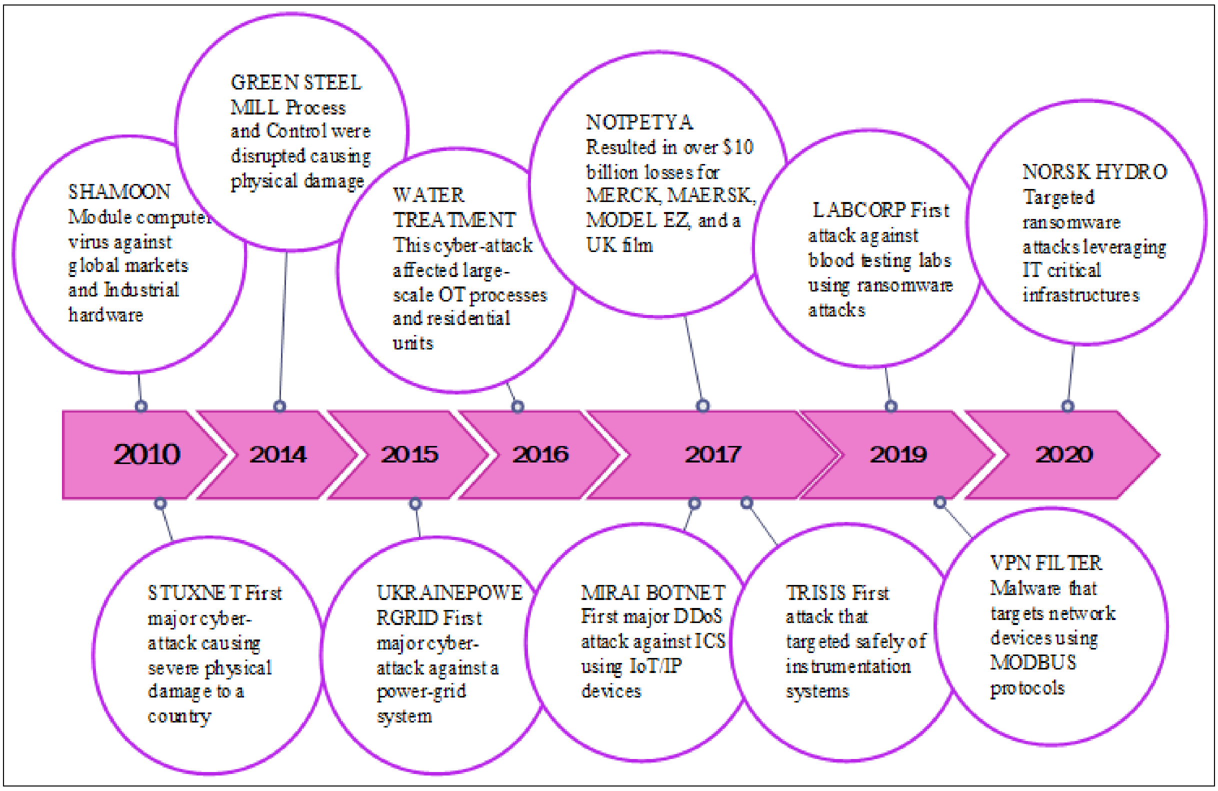

GREEN STEEL MILL Process and Control were disrupted causing physical damage
SHAMOON Module computer virus against global markets and Industrial hardware
WATER TREATMENT This cyber-attack affected large-scale OT processes and residential units
NOTPETYA Resulted in over $10 billion losses for MERCK, MAERSK, MODEL EZ, and a UK film
LABCORP First attack against blood testing labs using ransomware attacks
NORSK HYDRO Targeted ransomware attacks leveraging IT critical infrastructures
2010
2014
2015
2016
2017
2019
2020
STUXNET First major cyber-attack causing severe physical damage to a country
UKRAINEPOWERGRID First major cyber-attack against a power-grid system
MIRAI BOTNET First major DDoS attack against ICS using IoT/IP devices
TRISIS First attack that targeted safely of instrumentation systems
VPN FILTER Malware that targets network devices using MODBUS protocols

Fig 1: History of cyber assaults on industrial control systems during the last decade

Between 2016 and 2017, a breach of Shamoon's first version was used in conjunction with the Stone Drill malware in a series of large cyber-attacks [12]. Another big cyber-attack occurred in 2018 when three US gas pipeline companies stated that their digital communication networks had been knocked down for many days [13]. Cyber-attacks on Saipem, an Italian oil and gas company, were reported in October of the year [12]. Saipem's infrastructure in the Mideast, India, Scotland, and Italy were allegedly attacked. Symantec, in response to Saipem's statement, revealed evidence of similar assaults on two other Saudi Arabian and UAE-based fossil fuel corporations, as well as Filename malware [30]. According to Kaspersky Lab, malicious cyber activities and security incidents against commercial vital infrastructure facilities have increased by more than 50% in the recent decade [14, 15].

The goal of this thesis is to develop a deep learning-based system that can detect, categorizing, and locating the location of cyber-attacks against ICS and APT. It is essential to do an analysis of the malware and get its artifacts so that we can acquire a better idea of who the attacker is. The passage of time and the growing reliance of our society on our technical infrastructure will only serve to increase the frequency with which cyber assaults are carried out. The classification of malware according to kind, authorship, and other important categories will be facilitated with the assistance of cyber-attack attribution, which will be an essential component in the fight against the creation of malware. This is particularly helpful when thinking about malware that is sponsored by the state. Due to the fact that conventional forms of warfare will eventually be replaced by cyber-warfare, it is becoming more important to be able to determine who created the malicious software as well as the nation from which it originated. The members of our team are fluent in Python and have prior experience working with machine learning methods.

2. Materials and methods on ensemble deep learning to detect cyber-attacks

The original contribution's last section analyzes an ensemble deep learning-based model for attack detection in heterogeneous datasets derived from industrial machinery control systems. Later, it delves into how to identify the source of cyber-attacks in both small and large-scale manufacturing plants with the use of self-tuning, scalable, deep learning classification models. Here, we'll detail the process and rationale behind each contribution and show how they satisfy the stated goals.

In the first experiment, several diverse real-world ICS datasets are used to evaluate a generalized ensemble deep learning approach to cyber-attack detection in ICS. We suggest using this method in the pilot research. Multiple unsupervised RF, each learning unique representations from unbalanced datasets, make up the proposed deep learning model. Next, the representations produced by each RF are passed to a Deep Neural Network (DNN) through a super vector, where they are fused using a fusion activation vector. Finally, a Decision Tree (also known as a binary classifier) is utilized to identify hazards from the rearranged representations, rather than a traditional classifier.

- Making a learning model for deep representations with the intention of constructing new representations that are fairer. Resilience (as measured by f-score) and attack detection accuracy (both also improved) in an imbalanced setting were both a result of the new representations.
- Second, employing a deep learning ensemble built on RF classifiers to recognize cyberattacks based on the new representations, which improves detection accuracy while concurrently decreasing false positives.
- Third, develop a generic model that requires little to no adaptation to work across a wide range of critical infrastructure facilities. To detect cyberattacks in novel ICS settings, the proposed architecture employs easily trainable self-learning methods.

Data Collection

Initially, the suggested framework's performance was measured against that of randomly generated ICS and APT models using two distinct ICS datasets. Both come from separate sources; the former is sourced through a network of gas pipelines, while the latter is processed in a water purification plant. One ICS dataset was chosen from a water distribution plant and another from an electric power intelligent control system to help develop and update the final framework's ability to pinpoint the precise site of cyber-attacks.

Developing Initial Framework

The first step in testing the effectiveness of the proposed framework was to compare it against a random ICS model using two separate ICS datasets and APT for another database. The first is taken via a gas pipeline system, while the second is gathered at a water treatment facility. With the goal of refining and enhancing the overall architecture, we decided to include data from a water distribution facility and a smart control system for electric power into the ICS. The goal of these data sets is to facilitate research on the physical locations of cyber-attacks.

Final Framework

While this is a great way to boost the f-score of your classifiers, the examination of the first application shows that pinpointing the exact site of an attack without jeopardizing an industrial process is quite challenging. This is true despite the fact that this strategy produces strong results. Because of this, we devised scalable deep learning and categorization models that can substantially identify cyber-attacks, reduce the amount of downtime experienced after a breach has been discovered, and limit the downstream damage to equipment.

2.1 Ensemble Deep Learning-based Cyber-Attack Detection Algorithm for IICSD

This section introduces a flexible computational intelligence approach that can process raw, imbalanced data. With this model, we hope to avoid some of the pitfalls of earlier approaches. To improve the accuracy of a classification task using a deep learning ensemble model [16, 17], we first construct a new, more balanced representation of the data from the original dataset. Each instance of the deep learning model is an unsupervised RF that acquires its own unique representations from data that is not uniformly distributed. In order to identify attacks, the RF model employs several auto-encoders (AE) to learn new representations from unlabeled input and thereby gain diverse patterns. Then, a DNN is given a super vector including the freshly produced representations from each RF [18], and the representations are fused using the activation vector from the super vector. Last but not least, we utilize a DT in the form of a binary classifier to identify assaults using the reworked representations. Figure 2 displays the overall framework of the new model that is being presented.

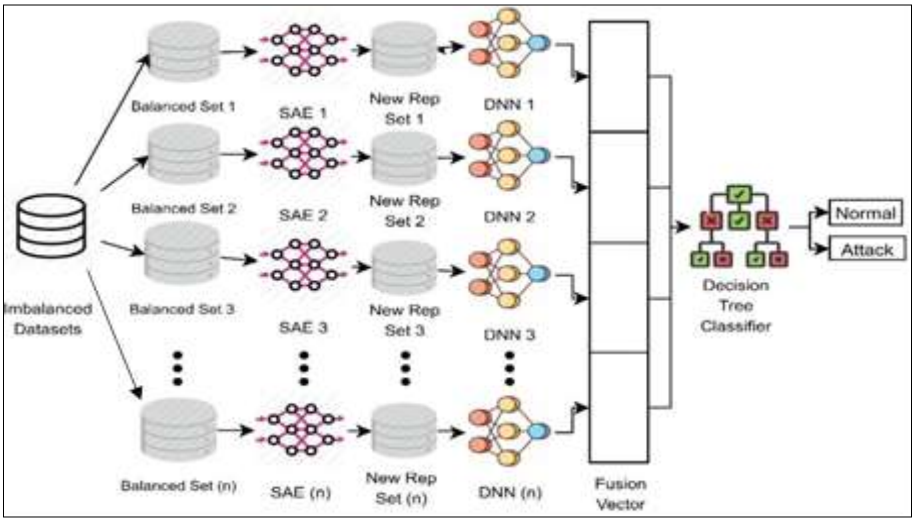


Fig 2: An orientation to the stacked auto-encoder concept and method

2.2 The Proposed Ensemble Deep Representation Learning Model

The vast inequalities that exist in real-world ICS are mostly disregarded by both existing methodologies and those offered in the literature (the number of attack samples are a lot less than the number of normal samples). This will result in weak f-measures, which are indicative of how poorly these models perform in an imbalanced setting such as ICS. Therefore, due to their poor performance in such settings, these models cannot be used in practical applications.

Misclassification of the newly revealed hazardous data is quite likely once a model has been deliberately trained using an exceedingly imbalanced dataset. To address this challenge, we developed an RF-based ensemble deep representation-learning model. The overall efficiency of the model will increase thanks to this modification. In order to develop new representations from the data, an equal balanced set is extracted and then passed on to several AE [19]. The input sample xi from the hidden layer's corresponding sample set X is represented by the equation below.

$$h_i = f(x_i) = \sigma(m_1 x_i + c_1) \quad (1)$$

Where M and c stand for the neuron weight matrices and bias vectors of the input and hidden layers, respectively [20, 21]. Once training has commenced, stacked multi-layer AEs may be built by using the function of the hidden layer to update the subsequent input layer. Despite the fact that using an ensemble model might improve computing efficiency somewhat, it was evident that using many AE would result in far superior f-measure scores.

Our model's generalizability is improved by adding a dropout layer, which makes the final result less sensitive to the original input. Multi-network cross-validation and in-depth evaluation of past loss experience and accuracy led to the optimal choice of nodes and layers. In order to express the cost function, which is based on binary cross entropy (BCE), we use the following notation:

$$J = -\frac{1}{N}\sum_{i=1}^{N} y_i . log(p(y_i)) + (1 - y_i). log\,(1 - p(y_i)) \quad (2)$$

Both the attack and control samples are denoted by the variables y1 and y2 here. The likelihood of discovering an attack sample, p(y), may be larger than zero. To prevent the AE's hidden layer from shrinking and the system from becoming slow and unlearning, BCE was used instead of MSE.

2.3 Model for Detecting Attacks Using an Ensemble of Deep Learning Techniques

Once the new models are built using the imbalanced dataset, an ensemble of DNN classifiers is utilized to differentiate between typical and aberrant behavior. After the representations are fused, a super vector with a fusion activation function is generated and fed into a DT classifier to be used in the detection of attacks. The DT [23] classifier was selected as the most efficient after extensive testing of other machine learning classifiers. The activation function for fusion in the sigmoid layer may be written as follows:

$$L_l = -\frac{1}{N}\sum_{i=1}^{m} y_i . \log(t_i). w_s + (1 - y_i). log\,(1 - t_i).w_l \quad (3)$$

When y_i is the label of the ith sample and ti is the prediction of the ith sample, L1 is the fusion perceptron of the sigmoid layer. Samples that are considered unstable have a weight of ws, whereas samples that are considered stable have a weight of wl. To better discover unstable samples, we make ws greater than wl and use wl = 1 as a baseline for successful unstable pattern mining [24, 26].

The accuracy and f-measure were increased by experimenting with different numbers of hidden layers, networks, batch sizes, training methods, epochs, and dropout layers in a for-loop. Rectified Linear Unit (ReLU) activation function is used in both RF and DNN [27] for maximum effectiveness on the following metrics:

$$ReLU\,(x) = \max(0, x) \quad (4)$$

where x is your own personal experience. Approach 1 provides a concise explanation of the suggested algorithm for the detection of attacks.

3. Data Preprocessing

It was then necessary to sanitize the data for usage in the machine learning model. All values in the language and library columns have been converted to uppercase to prevent the model from incorrectly distinguishing between them. The library and language features columns were then one-hot encoded [28]. As a consequence, it became necessary to set up fake variables to represent the various library and language combinations. Finally, the APT categories were converted into numbers for use in the classification algorithm. Another 792 rows were deleted because they included numerous columns with null values. It was reduced to 148 characteristics and 2862 rows after preprocessing.

3.1 Data Classification

We needed to think about whether or not the malwares were packaged so that we could do static analysis. Because the malware's format has been altered by packing, a sort of code obfuscation, by compression or encryption, we would require special unpacking tools in order to get the necessary artifacts from the program. Thus, we separated all of our malware into unpacked and packed categories using a program called PEiD that can tell whether a sample of malware is packed. The date provides a summary of the total number of both packed and unpacked malware samples across all APT families [29]. For attribution purposes, we have only ever utilized unpacked malware samples. There are 3,591 malware samples once they are decompressed.

We developed a python script that would gather all of the aforementioned information into a single csv file after first extracting it from each of the json malware report files. We prepared a CSV file that included the aforementioned three properties, resource (also known as the hash of the malware), and APT group columns, and then we downloaded it from the GitHub repository.

4. Results and Experiments

4.1 APT Malware Dataset

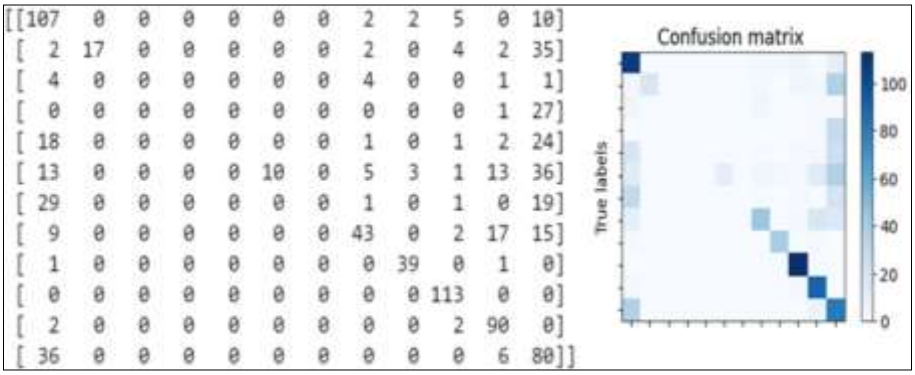
About 3,740 unique malware strains have been linked to 13 Advanced Persistent Threat (APT) groups that, according to reports, get financing from at least five different countries. Several Machine Learning algorithms were run on this dataset to determine authorship, and the results were compared using this data as a baseline. Future benchmarks or malicious software might benefit from this dataset's analysis.

The usage of several decision trees in Random Forest (RF) helps to minimize over-fitting, thus we choose to employ it for our machine learning investigation. The crucial aspects of Random Forest are also relatively simple to compute. Additionally, the Random Forest prevents error correlation among trees used for making predictions. For this reason, we decided to divide the total amount of data into two equal halves for use in training and testing. The RF Classifier in the scikit-learn library was put to good use. Initial values for the RF Classifier's hyperparameters were: • min samples leaf = 50, min_samples_leaf = 50, n_estimators = 150, bootstrap = True, oob_score = True, n_jobs = -1, random_state = seed, Max features = 'auto' 58% accuracy was the best performance for the model. Figure 3, Figure 4, and Table 1 show the confusion matrix, precision, recall, and f1-score assessment results obtained prior to preprocessing the data, respectively.

We have evaluated the model under a variety of unbalanced conditions to see how well the suggested strategy performs. An unbalanced ratio of 0.1 indicates 10% attack and 90% normal samples, while an imbalanced ratio of 1 indicates 50% attack and 50% normal samples.

Table 1: Performance evaluation before preprocessing malware data

Classes	Precision	Recall	F1-score	support
C1	0.48	0.85	0.62	126
C2	1.00	0.27	0.43	62
C3	0.00	0.00	0.00	10
C4	0.00	0.00	0.00	28
C5	0.00	0.00	0.00	46
C6	1.00	0.12	0.22	81
C7	0.00	0.00	0.00	50
C8	0.74	0.50	0.60	86
C9	0.89	0.95	0.92	41
C10	0.88	1.00	0.93	113
C11	0.68	0.96	0.79	94
C12	0.32	0.66	0.43	122
Accuracy			0.58	859



```
[[107   0   0   0   0   0   0   2   2   5   0  10]
 [  2  17   0   0   0   0   0   2   0   4   2  35]
 [  4   0   0   0   0   0   0   4   0   0   1   1]
 [  0   0   0   0   0   0   0   0   0   0   1  27]
 [ 18   0   0   0   0   0   0   1   0   1   2  24]
 [ 13   0   0   0   0  10   0   5   3   1  13  36]
 [ 29   0   0   0   0   0   0   1   0   1   0  19]
 [  9   0   0   0   0   0   0  43   0   2  17  15]
 [  1   0   0   0   0   0   0   0  39   0   1   0]
 [  0   0   0   0   0   0   0   0   0 113   0   0]
 [  2   0   0   0   0   0   0   0   0   2  90   0]
 [ 36   0   0   0   0   0   0   0   0   0   6  80]]
```

Fig 3: Confusion matrix before preprocessing malware data

Form the results above on data before preprocessing, we observe the accuracy is 58% on data supported 859 for 12 classes shown in Table 3.

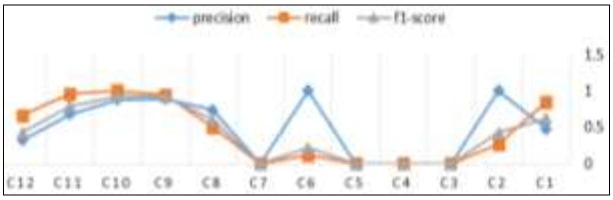
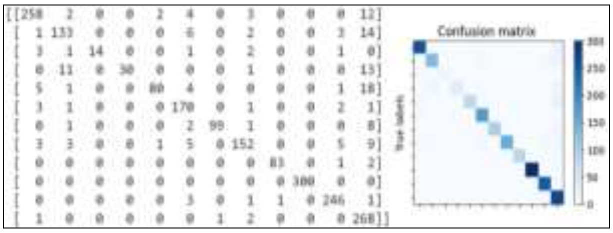


Fig 4: Classification performance on different classes before malware data preprocessing (C indicates to class)

In order to fine-tune the system, we adjusted the hyperparameters. To get a more accurate reading, we lowered the minimum number of estimators to 300 and the number of leaf samples to 1. Roughly 83% of predictions were correct. Furthermore, we use random cross-validation to check for over- and under-fitting in our model. As shown by 20-fold random cross-validation, our model is neither under fitting nor over fitting the data, as we reach an accuracy of 92% and 99%, respectively, using 1 and 3 leaf samples, respectively. Moreover, we determined which characteristics were most important to the categorization process. (Figs. 5, 6, 7, and 8; Tabs. 3 and 4).

Table 3: Performance evaluation on preprocessed malware data (No of leaf samples is 3)

Classes	Precision	Recall	F1-score	Support
C1	0.94	0.92	0.93	281
C2	0.87	0.84	0.85	159
C3	1.00	0.64	0.78	22
C4	1.00	0.55	0.71	55
C5	0.96	0.73	0.83	109
C6	0.87	0.96	0.91	178
C7	0.99	0.89	0.94	111
C8	0.92	0.85	0.89	178
C9	0.99	0.97	0.98	86
C10	1.00	1.00	1.00	300
C11	0.95	0.98	0.96	252
C12	0.77	0.99	0.87	272
Accuracy			0.92	2003



```
[[258   2   0   0   2   4   0   3   0   0   0  12]
 [  1 133   0   0   0   6   0   2   0   0   3  14]
 [  3   1  14   0   0   1   0   2   0   0   1   0]
 [  0  11   0  30   0   0   0   1   0   0   0  13]
 [  5   1   0   0  80   4   0   0   0   0   1  18]
 [  3   1   0   0   0 170   0   1   0   0   2   1]
 [  0   1   0   0   0   2  99   1   0   0   0   8]
 [  3   3   0   0   1   5   0 152   0   0   5   9]
 [  0   0   0   0   0   0   0   0  83   0   1   2]
 [  0   0   0   0   0   0   0   0   0 300   0   0]
 [  0   0   0   0   0   3   0   1   1   0 246   1]
 [  1   0   0   0   0   0   1   2   0   0   0 268]]
```

Fig 5: Confusion matrix on preprocessed malware data (No of leaf samples is 3)

The value for the pe-resource-langs property, neutral, is second only in importance to the connection point attribute.

Form the results above on preprocessing data we observe the accuracy is 92% on data supported 2003 for 12 classes using number of leaf samples is 3 shown in Table 3.

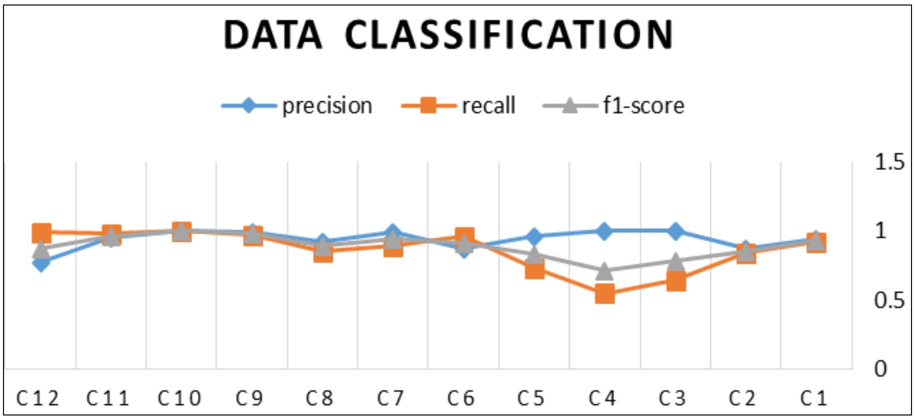
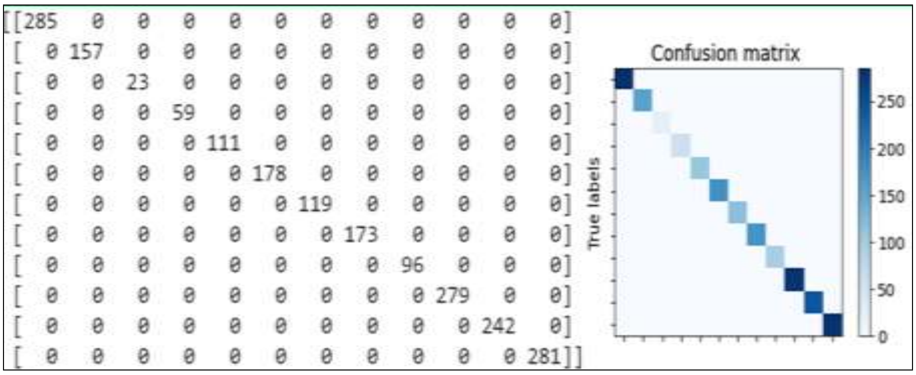


Fig 6: Classification performance on different classes on malware data preprocessed (No. of leaf samples is 3)



```
[[285   0   0   0   0   0   0   0   0   0   0   0]
 [  0 157   0   0   0   0   0   0   0   0   0   0]
 [  0   0  23   0   0   0   0   0   0   0   0   0]
 [  0   0   0  59   0   0   0   0   0   0   0   0]
 [  0   0   0   0 111   0   0   0   0   0   0   0]
 [  0   0   0   0   0 178   0   0   0   0   0   0]
 [  0   0   0   0   0   0 119   0   0   0   0   0]
 [  0   0   0   0   0   0   0 173   0   0   0   0]
 [  0   0   0   0   0   0   0   0  96   0   0   0]
 [  0   0   0   0   0   0   0   0   0 279   0   0]
 [  0   0   0   0   0   0   0   0   0   0 242   0]
 [  0   0   0   0   0   0   0   0   0   0   0 281]]
```

Fig 7: Confusion matrix on preprocessed malware data (No. of leaf samples is 1)

Table 4: Performance evaluation on preprocessed malware data (No. of leaf samples is 1)

Classes	Precision	Recall	F1-score	Support
C1	1.00	0.99	1.00	285
C2	1.00	1.00	0.99	157
C3	0.99	0.99	1.00	23
C4	1.00	1.00	1.00	59
C5	0.99	1.00	1.00	111
C6	1.00	1.00	0.99	178
C7	1.00	1.00	0.99	119
C8	1.00	0.99	1.00	173
C9	0.99	1.00	1.00	96
C10	1.00	1.00	0.99	279
C11	1.00	0.99	1.00	242
C12	1.00	1.00	0.99	281
Accuracy			0.99	2003

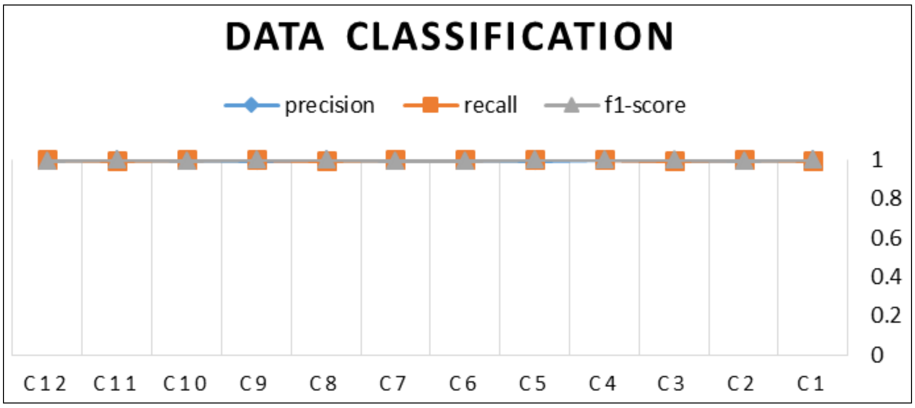


Fig 8: Classification performance on different classes on malware data preprocessed (No. of leaf samples is 1)

Form the results above on malware preprocessed data we achieved the accuracy is 99% on data supported 2003 for 12 classes using number of leaf samples is 1 shown in Table 5.

Algorithm for Imbalanced ICS Datasets

Two ICS datasets, one from a gas pipeline system and one from a water treatment plant, are used to assess the effectiveness of the suggested technique. RF Classifiers and a few additional approaches from the literature were used to evaluate the outcomes. Accuracy, precision, recall, and f-measures are only few of the assessment metrics summarized in Table 5. The dataset show that the proposed approach outperforms the state-of-the-art methods on all four measures, with the f-measure being particularly relevant in an unbalanced setting.

Table 5: Performance evaluation the detection capabilities using swat dataset

Model	Accuracy	Pre	Recall	F1-Score
Proposed RF	98.67	0.97621	0.98618	0.97619
[25] SVM	-	0.93521	0.70152	0.80002
[8] RNN	-	0.92544	0.69699	0.80563
[16] ID-CNN	-	0.95985	0.80251	0.88521
[32] TABOR	95.00	0.87233	0.79852	0.83655
[30] AE	-	0.89972	0.80787	0.84528

Table 5 indicate that the proposed RF model with data pretreatment outperforms other approaches in performance assessment detection on the various datasets. Here we discuss how each contribution fared in the review process. In our initial contribution, we use three datasets collected from various critical infrastructure sites to assess the efficacy of the suggested RF technique. The suggested extended models achieved 10% better f1-score, accuracy, recall, and precision than various peer methods in the current literature and conventional classifiers. Not to mention the fact that we can now identify attacks in ICS and APT.

With just three primary characteristics, 2862 malware samples, and two ICS datasets, the model performed quite well. Incorporating other characteristics, such entropy, the number of sections, or more malware samples, might improve the model's accuracy. Due to time constraints and a shortage of APT malware datasets, we extracted fewer characteristics. As an added downside, the time and effort spent on data preparation was quite extensive. More data may be added to the collection in the future by extracting more properties from the VirusTotal report. Use of neural networks for categorization and dynamic analysis in the Cuckoo sandbox are also viable options.

Conclusion

Reliable and secure operations of critical infrastructures are crucial to national security, since these systems constitute the backbone of contemporary civilization and include a wide range of cyber and physical systems. As a first contribution, we aimed to develop a cyber-attack detection technique (RF) for ICS that relies on generalized ensembles of deep learning models. A deep representation-learning model is used in the proposed method to create new balanced representations from the original unbalanced dataset. After the new representations have been created, a DNN and DT classifier-based ensemble deep learning technique is utilized to identify cyber-attacks. Two independent ICS datasets culled from operational facilities in the critical infrastructure sector are used to validate the proposed model's performance. Both the Gas Pipeline dataset and the Secure Water Treatment dataset saw improvements in accuracy thanks to our suggested method, with 95.00 percent for the former and 98.67 percent for the latter. The results were compared to those obtained by using more conventional classifiers like DNN and ADA, as well as by using other methods given by experts in the field. In each of the four measures used to compare the methods, the suggested methodology came out on top.

The accuracy and f1-score outcomes of our suggested method were 10% higher than those of traditional classifiers. Our generalized model may be easily applied to larger scale models, such as the water treatment system, and used in a variety of infrastructure facilities with few modifications to preexisting models. Additionally, 10-fold cross validation was used to assess the process and component models' correctness and f1-score by testing each data point just once and removing the possibility of bias from utilizing duplicate data points.

References

1. Higgins KJ. Security incidents rise in industrial control systems. Security Dark; c2010.
2. Al-Abassi A, Karimipour H, Dehghantanha A, Parizi RM. An ensemble deep learning-based cyber-attack detection in the industrial control system. IEEE Access. 2020;8:83965-83973.
3. Hadziosmanovic D, Bolzoni D, Etalle S, Hartel P. Challenges and opportunities in securing industrial control systems. In: 2012 Complexity in Engineering (COMPETING). Proceedings. IEEE; c2012. p. 1-6.
4. Sakhnini J, Karimipour H, Dehghantanha A, Parizi RM, Srivastava G. Security aspects of Internet of Things aided smart grids: A bibliometric survey. Internet of things. 2021;14:100111.
5. Karimipour H, Dinavahi V. Extended Kalman filter-based parallel dynamic state estimation. IEEE transactions on smart grid. 2015;6(3):1539-1549.
6. Ghalavand F, Alizade B, Gaber H, Karimipour H. Microgrid islanding detection based on mathematical morphology. Energies. 2018;11(10):2696.
7. Karimipour H, Leung H. Relaxation-based anomaly detection in cyber-physical systems using ensemble Kalman filter. IET Cyper-Phys. Syst.: Theory & Appl. 2020;5(1):49-58.
8. Tahsien SM, Karimipour H, Spachos P. Machine learning-based solutions for the security of Internet of Things (IoT): A survey. Journal of Network and Computer Applications. 2020;161:102630.
9. Al-Abassi A. Application of Deep Learning on Cyber-Attack Detection and Identification in Industrial Control Systems [Doctoral dissertation]. University of Guelph; c2020.
10. Zhang F, Kodituwakku HADE, Hines JW, Coble J. Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. IEEE Transactions on Industrial Informatics. 2019;15(7):4362-4369.
11. Langner R. Stuxnet: Dissecting a cyber-warfare weapon. IEEE Security & Privacy. 2011;9(3):49-51.
12. Ashok A, Edgar T. A High-Fidelity Cyber-Physical Testbed-Based Benchmarking Dataset for Testing

Operational Technology Specific Intrusion Detection Systems. In: 2021 IEEE International Symposium on Technologies for Homeland Security (HST). IEEE; c2021. p. 1-7.

13. Kalashnikov A, Sakrutina E. The model of evaluating the risk potential for critical infrastructure plants of nuclear power plants. In: 2018 Eleventh International Conference" Management of large-scale system development"(MLSD). IEEE; c2018. p. 1-4.
14. Kaspersky IC S. The threat landscape for industrial automation systems; c1997.
15. Lu G, Feng D. Network security situation awareness for industrial control system under integrity attacks. In: 2018 21st International Conference on Information Fusion (FUSION). IEEE; c2018. p. 1808-1815.
16. Al-Abassi A. Application of Deep Learning on Cyber-Attack Detection and Identification in Industrial Control Systems [Doctoral dissertation]. University of Guelph; c2020.
17. Fovino IN, Carcano A, Murel TDL, Trombetta A, Masera M. Modbus/DNP3 state-based intrusion detection system. In: 2010 24th IEEE International Conference on Advanced Information Networking and Applications. IEEE; c2010. p. 729-736.
18. Kang B, McLaughlin K, Sezer S. Towards a stateful analysis framework for smart grid network intrusion detection. In: 4th International Symposium for ICS & SCADA Cyber Security Research 2016; c2016. p. 124-131.
19. Stouffer K, Pillitteri V, Lightman S, Abrams M, Hahn A. Guide to Industrial Control Systems (ICS) Security Supervisory Control and Data Acquisition (SCADA) Systems, Distributed Control Systems (DCS), and Other Control System Configurations such as Programmable Logic Controllers (PLC); c2015.
20. Abdulrazzaq M, Wei Y. Industrial Control System (ICS) Network Asset Identification and Risk Management; c2018.
21. Darabian H, Homayounoot S, Dehghantanha A, Hashemi S, Karimipour H, Parizi R, *et al*. Detecting Cryptomining Malware: a Deep Learning Approach for Static and Dynamic Analysis. Journal of Grid Computing; c2020.
22. Hixon R, Gruenbacher DM. Markov chains in network intrusion detection. In: Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop. IEEE; c2004.
23. Lv F, Han M, Qiu T. Remote Sensing Image Classification Based on Ensemble Extreme Learning Machine with Stacked Autoencoder. IEEE Access. 2017;5:9021-9031.
24. Pecht MG, Kang M. Machine learning: Anomaly detection; c2019.
25. Anton SD, Sinha S, Schotten HD. Anomaly-based intrusion detection in industrial data with SVM and random forests. In: 2019 27th. International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2019; c2019.
26. Alves T, Das R, Morris T. Embedding Encryption and Machine Learning Intrusion Prevention Systems on Programmable Logic Controllers. IEEE Embedded Systems Letters. 2018;10(3):99-102.
27. Syed ZAS. Anomaly Detection with Machine Learning in Wireless Networks and IoT [Master's thesis]; c2021.
28. Linda O, Vollmer T, Manic M. Neural Network based intrusion detection system for critical infrastructures. In: Proceedings of the International Joint Conference on Neural Networks; c2009.
29. Yang H, Cheng L, Chuah MC. Deep-Learning-Based Network Intrusion Detection for SCADA Systems. In: 2019 IEEE Conference on Communications and Network Security (CNS); c2019.
30. Beaver JM, Borges-Hink RC, Buckner MA. An evaluation of machine learning methods to detect malicious SCADA communications. In: Proceedings - 2013 12th. International Conference on Machine Learning and Applications, ICMLA 2013; c2013.
31. Shirazi SN, Gouglidis A, Syeda KN, Simpson S, Mauthe A, Stephanakis IM, *et al*. Evaluation of anomaly detection techniques for SCADA communication resilience. In: Proceedings - 2016 Resilience Week, RWS 2016; c2016.
32. Lin Q, Verwer S, Adepu S, Mathur A. TABOR: A graphical model-based approach for anomaly detection in industrial control systems. In: ASIACCS - Proceedings of the ACM Asia Conference on Computer and Communications Security; c2018.