IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 10, 2021, accepted August 29, 2021, date of publication September 7, 2021, date of current version September 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3111053

Minority Resampling Boosted Unsupervised Learning With Hyperdimensional Computing for Threat Detection at the Edge of Internet of Things

VIVEK CHRISTOPHER[1], THARMASANTHIRAN AATHMAN[1], KAYATHIRI MAHENDRAKUMARAN[1], RASHMIKA NAWARATNE[2], DASWIN DE SILVA[2], (Senior Member, IEEE), VISHAKA NANAYAKKARA[1], AND DAMMINDA ALAHAKOON[2], (Member, IEEE)

[1]Department of Computer Science and Engineering, University of Moratuwa, Moratuwa 10400, Sri Lanka
[2]Research Centre for Data Analytics and Cognition, La Trobe University, Melbourne, VIC 3086, Australia

Corresponding author: Daswin De Silva (d.desilva@latrobe.edu.au)

ABSTRACT The Internet of Things (IoT) has rapidly transformed digital environments across a multitude of domains with increased connectivity and pervasive virtualization. The distributed computing paradigm of Edge Computing has been postulated to overcome the concerns of response time, bandwidth, energy consumption, and cybersecurity. In comparison to the other concerns, limited studies have focused on cybersecurity, mainly due to the inherent complexity of threat detection at the Edge. However, the widespread adoption of IoT applications in economic, social, and political contexts is a stringent indication of the significant impact from cyber-attacks. This paper aims to address this challenge by presenting an effective and efficient machine learning approach for threat detection at the Edge of IoT. The novel contributions of this approach are, a new Enhanced Geometric Synthetic Minority Oversampling Technique (EG-SMOTE) algorithm to resolve the imbalanced distribution of data streams at the IoT Edge, an extension to the Growing Self Organizing Map (GSOM) algorithm based on Hyperdimensional Computing for energy efficient machine learning from unlabeled data streams. The proposed EG-SMOTE + GSOM approach has been tested using four open access datasets; three benchmark, KDD99 (F-Score = 0.9360), NSL-KDD (F-Score = 0.9647), CICIDS2017 (F-Score = 0.9999), and one industry-focused botnet IoT traffic dataset, BoT-IoT (F-Score = 0.9445). The EG-SMOTE approach has outperformed SMOTE and G-SMOTE approaches in a vast number of experiments that are tried with different classifiers. The results of these experiments confirm the novelty, efficiency and effectiveness of this approach for cybersecurity at the IoT Edge.

INDEX TERMS Edge computing, cybersecurity, edge IoT, hyperdimensional computing, minority resampling unsupervised machine learning, growing self organizing map algorithm.

I. INTRODUCTION

Edge Computing is primed to address the challenges of Internet of Things applications that are being developed and deployed in complex real-world settings [1]. The Internet of Things Edge (IoT Edge) has enabled computation and storage in end-user proximity, decreased transmission latency, and reduced network bandwidth requirements, leading to efficiencies in response time, resource utilization and end-user outcomes [2], [3]. This has been particularly significant for real-time IoT Edge applications in energy management, smart factories, and digital healthcare [4], [5]. Despite these advances, there has been limited research conducted on effective, efficient and secure machine learning at the Edge of IoT [6]. Furthermore, a recent taxonomic analysis highlighted the importance of a trust ecosystem for cybersecurity in order to improve the uptake and proliferation of IoT Edge applications [7].

The IoT is a key enabling technology in Industry 4.0. Cybersecurity threats and attacks on IoT Edge applications have been categorized into three layers, perception, transportation and application [7], [8], and further studied in terms of class of attacks, key-related, denial of service, replay and privacy attacks [9]. A cybersecurity attack on an IoT

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaoqing Pan.

Edge application is a breach on the integrity of its structure, function, and operations, impacting both cyber and physical elements [10]. The types of cybersecurity attacks are diverse in terms of exploits, targets, methodologies, and the technical mechanism. These attacks aim to prevent the legitimate use of a service, compromise a user's security and privacy, interrupt system security and data integrity, gain compromised grant permissions and engineer malicious activities using DDoS attacks, side-channel attacks, malware injection attacks, authentication and authorization attacks, man-in-the-middle attacks, and bad-data injection attacks [11]. Encryption, key management and multi-factor authentication can be used to reduce these attacks [12], [13]. Each attack is usually intangible and can remain undetected for months, deteriorating the critical components of the IoT Edge. IoT-related vulnerabilities, if successfully exploited, can affect not only the device itself, but also the application field in which the IoT device operates [14]. Low computational capacities, protocol heterogeneities and coarse-grained access control [11], hardware and social engineering vulnerabilities [15] within an IOT Edge setting introduce further challenges for the detection of cyber threats and attacks. IDS (Intrusion Detection System) have been generally used to detect cyber-attacks in most application settings. IDS fall into two broad categories: signature-based and behaviour-based. Signature-based intrusion detection is based on pattern matching techniques to efficiently determine a known attack. Signature-based models require frequent updates with a new signature [15]. Behaviour-based intrusion detection, also known as anomaly detection, compares operational behavior profiles to detect attacks, based on deviations from profiles of normality. In IoT Edge, anomaly detection approaches are more effective than signature-based methods as most cyber physical attacks employ obfuscation techniques such as inserting no-ops, code re-ordering, register renaming, expanding and shrinking code, and the insertion of garbage code to bypass signature checks at databases [16], [17].

Current literature reports three types of approaches for anomaly detection. They are knowledge-based, statistical and machine learning approaches. Knowledge-based and statistical approaches are affected by the limitations of capturing, profiling and updating IoT Edge configurations at operation level in a dynamic computing environment, and the exposure of system vulnerabilities for behaviour profiling, whereas machine learning is able to address these limitations by managing the adaptive disposition and dynamic behavior of IoT Edge operations with high detection rates, low false positives and pragmatic computation and communication costs [18], [19]. More specifically, unsupervised machine learning methods are technically suited for the detection of behaviour-based cyber threat and attacks on IoT Edge as it can learn from unlabeled data [20]. In settings where machine learning is based on imbalanced datasets, the weakness of general learning algorithms contributes to the difficulties of classifying the anomalies as the algorithms generally bias towards the majority class samples. This limitation is more pronounced in IoT Edge where failing to account for minority data samples is consequential than the removal of such data due to underrepresentation.

Drawing on this context, we propose an effective, efficient and secure method for machine learning at the IoT Edge, specifically for cybersecurity threat detection. This method is effective as it addresses the challenge of high volume, high velocity unlabeled data streams generated at the IoT Edge. It is efficient as it conducts unsupervised machine learning using the Growing Self Organizing Map (GSOM) algorithm based on hyperdimensional computing, resulting in an energy-efficient computation and storage footprint. It is secure as it is boosted by minority resampling of imbalanced data generated by cybersecurity threats and attacks at the IoT Edge.

The following research contributions are reported in this paper.

- The development of a novel EG-SMOTE algorithm for resampling that addresses the limitations of synthesizing noisy minority samples, overfitting due to extreme synthesis of minority samples, and improper synthesis along the borderlines, specifically from imbalanced data streams in cybersecurity settings.
- A machine learning method that advances EG-SMOTE for unsupervised machine learning from unlabeled data in the IoT Edge. The unsupervised machine learning capability is based on the Growing Self Organizing Map (GSOM) algorithm that also utilizes Hyperdimensional Computing for energy-efficient machine learning.
- Empirical evaluation of the proposed approach using three benchmark datasets, KDD99, NSL-KDD, CICIDS2017, and an industry-focused botnet IoT traffic dataset, BoT-IoT that confirms the security, efficiency and effectiveness of the proposed machine learning approach at the IoT Edge.

The rest of this paper is organized as follows; Section II presents related work on threat detection in the IoT Edge, sampling imbalanced data, machine learning for threat detection, and hyperdimensional computing for low energy computation implementations. Section III delineates the proposed approach, focusing on the development of the new EG-SMOTE algorithm and its incorporation into the GSOM algorithm for unsupervised classification. Section IV reports on empirical evaluation that confirms the validity and effectiveness of the proposed approach. The paper concludes with Section V.

II. RELATED WORK

Threat detection in the IoT Edge can be generalized as deviations from standard behaviour of processes and functions within an IoT application. The original formulation of such deviations can be traced back to anomalies as defined by Hawkins, 'an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism' [21]. Anomaly detection can be used to detect cyber threats on the IoT Edge such as side-channel attacks, denial of service attacks, malware injection attacks and authentication attacks. The main goal in anomaly detection is to define a precise boundary between normal and anomaly data [22]. Numerous machine learning and traditional statistical approaches have been proposed for anomaly detection, however, only a few of these have been adopted for cyber threat detection in IoT applications [23]. An architecture for edge-based security, building firewalls, intrusion detection systems, authentication protocols and privacy preserving methods have been proposed recently [24], [25].

When the number of training data is insufficient, the few-shot learning (FSL) approach can be used which reduces the task related training via the prior knowledge. This is a learning paradigm that aims to address concerns with a shortage of training data by allowing models to identify novel categories with only a few sample data. The main drawback is that FSL requires a balanced dataset to detect the anomalies for intrusion detection [26]. The IoT anomaly datasets are usually imbalanced, where one class is represented by a large number of cases and the other is represented by only a few cases. Granular computing is an important technique for identifying the optimal granularity under an imbalanced dataset. Granular Computing (GrC) has risen to prominence as a new multi-disciplinary paradigm in artificial intelligence and has attracted a lot of attention in recent years. Xu *et al.* [27] states that GrC can be used as an efficient means of data-preprocessing step which is employed in many machine learning approaches. Many applications of GrC in the field of machine learning have recently been described by Ye *et al.* [28], and many outlier detection techniques based on GrC have also been proposed. GrC can be combined with deep learning techniques to identify minority patterns from imbalanced data for service planning.

The blockchain-based systems could help to prevent counterfeiting of data by ensuring that the IoT systems have not been tampered with. These systems are employed to overcome the serious concerns regarding the security in manufacturing and product lifecycle management in industry 4.0 such as, blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0; blockchain-secured smart manufacturing in industry 4.0; combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing [29]–[31]. Features of blockchain technology can be leveraged to provide an Anomaly Detection Service. NOKIA Bell Labs, proposed the first solution, Blockchain Anomaly Detection (BAD) for detecting anomalies in blockchain-based systems. Much study has been done on blockchain-enabled sustainable manufacturing in Industry 4.0 from a technical, commercial, organizational, and operational standpoint [32].

Most of the anomaly detection methods need human interactions [6]. Drawing on this limitation, the following subsections explore recent work related to the machine learning approach proposed in this paper.

A. RESAMPLING IMBALANCED DATA

Oversampling is the process of replicating the minority class and undersampling is the deletion of repeating samples of the majority class [33]. Extreme oversampling leads to overfitting despite the preservation of useful information and features, while undersampling leads to underfitting and poor generalization. SMOTE is an oversampling technique, which synthesizes new minority data along the line segments joining randomly chosen minority samples [34] By generating examples similar to existing minority points, SMOTE creates larger and less specific decision limits, which increase the generalization skills of classifiers and thus increase performance. Han *et al.* [35] suggested that synthetic samples must be created upon samples closed to the boundary and borderline-SMOTE algorithm is based on the sample's selection strategy. This borderline-SMOTE categorizes the minority instances as noise, safe, and danger sets. The data points in danger sets are considered as the borderline instances, and they are oversampled similar to SMOTE. Douzas and Bacao [36] proposed G-SMOTE for generating synthetic samples in a geometric region of the input space, around each selected minority instance. The basic configuration of this geometric region can be a hypersphere or a hyper-spheroid.

B. MACHINE LEARNING FOR ANOMALY DETECTION

Machine learning has proven to be far more effective than knowledge-based and statistical techniques for anomaly detection [37]. Existing literature reports all three types of machine learning for anomaly detection, supervised, unsupervised, and semi-supervised [20]. Supervised learning for anomaly detection is affected by the need for pre-labelled data of normal and anomalous behaviors. This is specifically challenging in an Iot Edge setting, where the data is inherently unlabeled, and the volume of anomalies can be large and not easily accessible or available from vendors or other end-users due to concerns of exposing further vulnerabilities. Given this is a significant limitation, it is pertinent to focus specifically on unsupervised learning techniques for threat detection. Eskin *et al.* [38] evaluated clustering, k-NN as well as a one-class SVM using KDD-Cup99 dataset.

Techniques like autoencoders are trained on normal data and can be used to detect anomalies [39]. In contrast to these unsupervised learning approaches, the GSOM algorithm transforms high-dimensional data into low-dimensional data while preserving the underlying topology representation of the data [40]. The GSOM algorithm has also been used for clustering, classification and visualization based on this property of dimensionality reduction. GSOM has been successfully adapted for DoS attack detection [41], and activity detection [42].

C. RESAMPLING IMBALANCED DATA

Hyperdimensional (HD) computing is a bio-inspired computational approach for representing and manipulating concepts and their meanings in a high-dimensional space with a low computational overhead [43]. High dimensional binary vectors of fixed length are the basis for representing information in this type of computing, and the information in an HD vector is evenly distributed across the vector's positions, therefore, hyperdimensional computing operates with distributed representations [44]. These distributed representations contrast with localist representations as they can be used to perform low-resource computations on digital acceleration hardware such as FPGA units available on IoT and Edge devices. Recent work [45]–[47], successfully demonstrated the effectiveness of the adaptation of the GSOM algorithm based on HD computing for unsupervised learning from unlabeled data in low energy devices and settings. As delineated in the following section, the proposed machine learning method expands on this success.

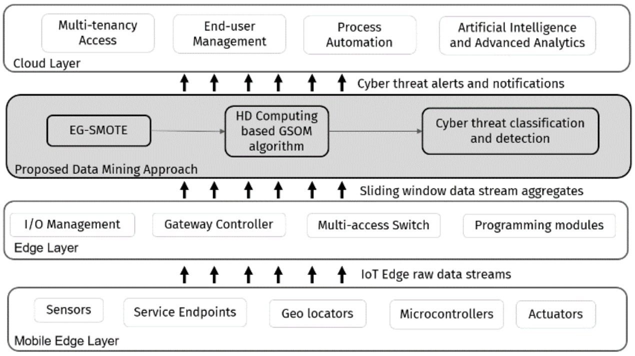
Multi-tenancy Access
End-user Management
Process Automation
Artificial Intelligence and Advanced Analytics
Cloud Layer
Cyber threat alerts and notifications
EG-SMOTE
HD Computing based GSOM algorithm
Cyber threat classification and detection
Proposed Data Mining Approach
Sliding window data stream aggregates
I/O Management
Gateway Controller
Multi-access Switch
Programming modules
Edge Layer
IoT Edge raw data streams
Sensors
Service Endpoints
Geo locators
Microcontrollers
Actuators
Mobile Edge Layer

FIGURE 1. Proposed machine learning approach in the context of the architecture for cloud-edge orchestration of IoT applications, based on [41] and [42].

III. THE PROPOSED APPROACH

We have designed the proposed machine learning approach in the context of the architecture for cloud-edge orchestration of IoT applications [48]–[50]. As illustrated in Fig. 1, it is positioned between the Cloud layer and the Edge layer as it receives data streams from IoT devices situated in both the Edge and Mobile Edge layers. This cooperative architecture across Cloud, Edge and Mobile Edge layers enables real-time responses for the detection of cyber threats and attacks. The proposed machine learning approach begins with the EG-SMOTE algorithm for resampling that generates balanced IoT data streams. Next, the adaptation of the GSOM algorithm based on HD computing performs unsupervised learning from unlabeled data, within the bounds of the computational constraints of the Edge layer. Finally, a classification module is used to identify anomalies and push these across as alerts and notifications to the Edge and Cloud layers.

A. THE DEVELOPMENT OF EG-SMOTE ALGORITHM

G-SMOTE extends the linear interpolation procedure by generating samples based on a geometric region [36]. This algorithm defines a geometric region around the specific sample inside which the new samples are synthesized.

Algorithm 1

```
Input: $S_{maj}$, $S_{min}$, N, k, $\alpha_{trunc}$, $\alpha_{def}$, $\gamma$, $\beta$
$S_{maj}$      - Samples of the majority class
$S_{min}$      - Samples of minority class
N              - Total number of synthetic samples to be
                 created
K              - k-nearest data points
$\beta$        - Sampling ratio (Smin: Smaj)
$\gamma$       -Sub-cluster resampling contribution
                 rate (SRCR)
ni             - Upper limit of synthetic samples that a
                 sub-cluster (ith sub-cluster) can generate
$\alpha_{trunk}$ - Truncation factor ($-1 \leq \alpha_{trunc} \leq 1$)
$\alpha_{def}$ - Deformation factor ($0 \leq \alpha_{def} \leq 1$)
n              - Number of sub-clusters
Output: S gen
Start
1. Define $\beta$, $\alpha_{trunc}$ and $\alpha_{def}$.
2. Calculate the number (N) of synthetic samples to be
   generated.
   $$N = Num\ (Majority) \left[ \frac{\beta}{1-\beta} \right] -Num\ (Minority) \quad (1)$$
3. Cluster the minority samples into an optimum number
   of sub-clusters (n).
4. Find the maximum number(ni) of synthetic samples
   that can be created by each cluster.
   $$n_i = (\gamma N) \frac{Number\ of\ minority\ points\ insub\ cluster}{Total\ number\ of\ minoritypoints} \quad (2)$$
5. Find k-nearest points $\forall$ x (x $\in S_{min}$)
   $L_{nearest}$    - list of k-nearest neighbors of x
   $L_{near\_min}$  - list of k-nearest minority neighbors of x
   $P_{maj}$        - nearest majority point of x
6. Execute the following steps until N number of synthetic points are generated.
   • Shuffle the minority points
   • Randomly choose a minority point $S_i \in S_{min}$
   • Selection Phase ($S_i$)
     Scenter = $S_i$
     If ($\forall$ y $\in L_{nearest}$) $\in S_{maj}$
         Continue
     Else if ($L_{nearest} = L_{near\_min}$)
         Modify $\alpha_{trunc}$
         Select a minority point $P_{min}$ ($P_{min} \in L_{near\_min}$)
         Ssurface = $P_{min}$
          point_generation ($S_{center}$, $S_{surface}$)
     Else
         Randomly select a minority point y
         (y $\in L_{near\_min}$)
         $S_{surface}$ = argmin $P_{min}$, $P_{maj}$ (|| $S_{center}$-$P_{min}$||,
         ||$S_{center}$-$P_{maj}$||)
```

Algorithm 1 *(Continue):*

```
        Point_generation ($S_{center}$, $S_{surface}$)
    **Point_generation ($S_{center}$, $S_{surface}$)**
    If $S_{surface} \in S_{min}$
        Modify $\alpha_{trunc}$
        Check the $n_i$ that $i^{th}$ cluster can generate s.t
        $S_i \in i^{th}$ cluster
        If ($n_i > 0$)
            Synthesize a point in a safe region
            $n_i = n_i - 1$
        Else
            Continue
        Else
            Modify $\alpha_{trunc}$
            Synthesize a point in a safe region
  • **Synthesize_sample ()**
```

EG-SMOTE extends G-SMOTE by implementing specific modifications in synthetic sample generation. The EG-SMOTE algorithm can be further discussed in six steps. Step 1 deals with the initialization process of the algorithm, defining and assigning values for the known parameters. These parameter inputs are given in the algorithm and will be elaborated in the following steps. Step 2 defines the number of samples to be synthesized. Step 3 and 4 are novel steps introduced into the algorithm to segregate the sub clusters to which the resampling can be re-applied. Step 5 and 6 are the critical part of the algorithm where the minority points are chosen at random and subjected to resampling based on the defined category. The above-mentioned steps are elaborated below.

Step 1: Truncation factor (α_{trunc}) and deformation factor (α_{def}) which were introduced in G-SMOTE [36] and Sampling rate (β), are initialized. Truncation factor corresponds to the transformation of the hyper-sphere into a hyper-spheroid. Similar to the truncation, the deformation transformation further modifies the initially uniform probability distribution. EG-SMOTE restricts the number of samples to be generated by sampling ratio β, in contrast to G-SMOTE which synthesizes new samples until the ratio of majority to minority becomes 1:1. Considering this 1:1 ratio of oversampling in G-SMOTE, the generation of humongous amounts of synthetic data will lead the models to learn unrealistic synthetic knowledge that do not exist in datasets. Therefore, EG-SMOTE restricts the rate of oversampling to reduce excessive synthesizing of minority samples. α_{trunc} and α_{def} are initialized with an initial value and later modified depending on the category of the selected minority sample (further explained in Step 6). In contrast, G-SMOTE operates with a static value for both factors.

Step 2: As mentioned above, the number of samples to be generated (N) is calculated using (1) based on the sampling ratio (β). Generally, N is calculated as the difference between the number of minority and majority samples which leads to the minority points to be synthesized in abundance such that minority, majority ratio be 1:1. This abundant creation of synthetic samples may lead to overfitting. Therefore, a parameter called sampling ratio (β) is introduced to limit the number of points to be generated.

Step 3 and 4: Minority samples are clustered into an optimum number of sub-clusters (n) to generalize the generation of new samples across all regions, which reduces the effect of overfitting. Since we employ an approach which allows resampling based on sub-clusters, sub-cluster resampling contributing rate γ ($0 < \gamma < 1$) is given a value such that the contribution from each sub-clusters in synthesizing new samples is constrained by an upper limit as presented in (2).

The k-means clustering algorithm is used to cluster the minority samples after obtaining optimum value for k from the 'Elbow Test' [51]. The number n_i limits the contribution to N from each sub-cluster to prevent overfitting and allow the synthesis of minority points from every other category.

Step 5: k-nearest points are identified for each minority sample in three categories: k-nearest points from minority samples, k-nearest points from both minority and majority samples and a nearest majority point.

Step 6: EG-SMOTE categorizes the selected minority point based on the k-nearest points and chooses the surface point based on the category where G-SMOTE fails to do so. Based on the ratio of (majority:minority) in k-nearest neighbors, minority samples are categorized. Consider m being the no. of majority samples in k-nearest neighbors.

- If $m = k$, it is an absolute noisy sample
- If $m >= 3k/4$, it is a noisy sample
    - ○ Both these samples are considered noisy algorithms. Borderline SMOTE has considered only the first case as noisy leaving the second as borderline sample [35]. But since there are possibilities for a miniature noisy cluster, with one to two minority points, EG-SMOTE refrains from treating those as borderlines and rather treat those as noisy. By this EG-SMOTE intends to prevent creation of more noisy samples.
- If $m = 0$, it is an absolute safe sample
- If $m <= k/4$, it is a safe sample
    - ○ EG-SMOTE reduces the threshold for safeness as opposed to Borderline SMOTE [35]. Borderline SMOTE never synthesizes new samples for safe zone data, which introduces an intra-cluster imbalance. EG-SMOTE algorithm addresses this intra-cluster imbalance, by synthesizing new samples for safe zone samples. But the algorithm shrinks the threshold for safe samples since extensive synthesis of minority data will lead to overfitting of data [52].
- ○ EG-SMOTE identifies a sample as borderline only when $k/4 < m < 3k/4$.

There is a necessity of addressing the inherent nature of minority samples. Hence, they are categorized accordingly, provided with different hyper-sphere selection phase, and point generation phase specific to that category. The point generation phase differs from one category to another by assigning different values for α_{trunc} and α_{def}. Hyper-sphere is pruned as per the category, and a new minority sample is synthesized. The method synthesis_sample for generating points follow similar steps of G-SMOTE. Points generation based on the above-mentioned categories is elaborated as follows.

There is a necessity of addressing the inherent nature of minority samples. Hence, they are categorized accordingly, provided with different hyper-sphere selection phase, and point generation phase specific to that category. The point generation phase differs from one category to another by assigning different values for α_{trunc} and α_{def}. Hyper-sphere is pruned as per the category, and a new minority sample is synthesized. The method synthesis_sample for generating points follows similar steps of G-SMOTE. Points generation based on the above-mentioned categories is elaborated as follows.

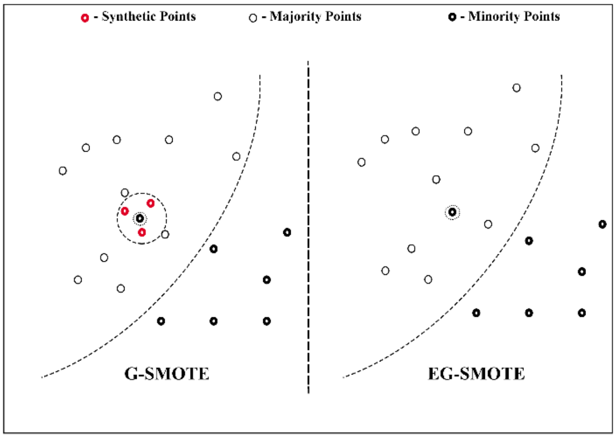

FIGURE 2. Incorporating samples for noisy instances.

1) NOISY SAMPLES

A minority sample is noisy when all the k-nearest neighbors are majority samples or when the majority is the most frequent (>80%) in the k-nearest neighbors. EG-SMOTE prohibits synthesizing new samples for noisy samples. G-SMOTE has identified the problem correctly, however, the algorithm does not prevent incorporating new samples from noisy minority samples. As a result, G-SMOTE has the potential to end up in synthesizing new instances for noisy samples, as depicted in Fig.2. Consider a scenario where all the k-nearest points belong to the majority, G-SMOTE selects the nearest majority sample as the surface point and tries to synthesize new minority instances in the hyper-sphere. Hence primarily, the G-SMOTE algorithm is enhanced to prevent the integration of further noisy minority samples.

2) BORDERLINE SAMPLES

Borderline minority samples occur when the existence of minority data in the k-nearest neighbors is above 40% but less than 80% as discussed above. These borderline samples are often located in overlapping regions of minority and majority classes or placed close to the complex decision boundaries between the types. It is essential to define a safe zone for point generation in the borderline of the minority and majority data clusters. This issue is proposed and handled by G-SMOTE correctly. However, G-SMOTE has a static truncation and deformation factor. The deformation factor deals with a plane of synthesis point where truncation deals with the pruning of the sphere to define a safe zone for point generation [36]. The negative values for the truncation factor would prune the same side of the selected surface point and vice versa. Consider a situation similar to Fig.3, where the surface point is a minority sample. G-SMOTE, with its truncation factor ($-1 <= \alpha_{\text{trunc}} <= 1$) being a static value (consider truncation factor to be 1.0), the algorithm prunes the same side of the selected surface point for both instances. This approach would be successful in some instances like when the surface point is a majority sample at the same time leading to synthesizing noisy samples when the minority point is selected to be the surface point.

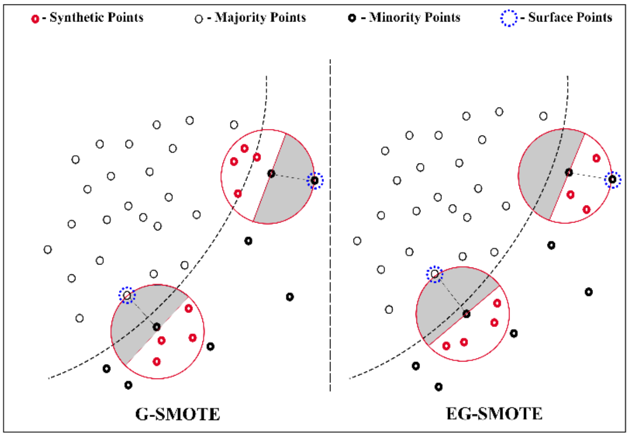

FIGURE 3. Synthesizing samples in borderline.

Based on the empirical evaluation, it was decided to define the truncation factor based on the point chosen as the surface point (either majority or minority) to reduce the impact of synthesizing minority data in the clusters of majority cluster space. Fig.3 shows the generation of new instances in the borderline between both binary clusters. With respect to EG-SMOTE, for a minority point, the truncation (α_{trunc}) factor is assigned to a value lesser than zero when the selected surface point belongs to the majority class such that pruning is done on the opposite side. If the surface point is the majority point, α_{trunc} will be assigned with a negative value from -1 to 0. Similarly, a positive value will be assigned when a minority sample is selected as the surface point where the opposite side will be pruned.

Safe zone sample is when almost all the k-nearest neighbors are minority samples. Borderline SMOTE [35] claims that sampling minority data leads to overfitting and discourages subcluster wise oversampling. G-SMOTE has evangelized about data synthesizing in vast spread areas or synthesizing data of different minority samples all over. This method is efficient in addressing the issues of imbalanced binary classification. As Bartosz Krawczyk addressed in [53], each minority sample should be considered for this interpretation. However, G-SMOTE has not considered the effect of over-fitting (Fig.4).

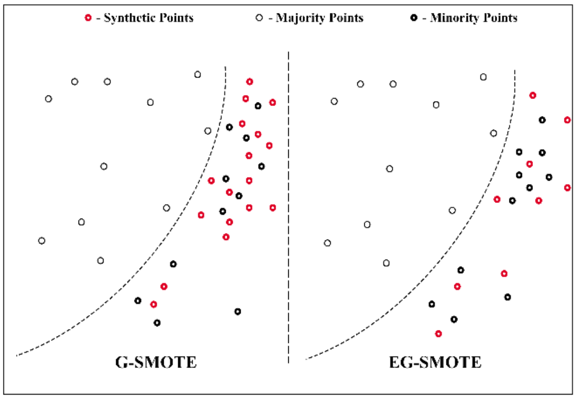

FIGURE 4. Synthesizing samples based on clusters.

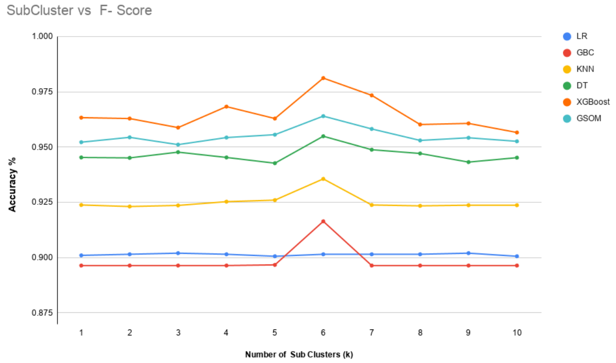

FIGURE 5. Elbow graph for number of clusters vs accuracy with different classifiers.

G-SMOTE arbitrarily allows minority points to be synthesized in abundance such that minority, majority ratio be 1:1. When considering a larger dataset, the number of synthetic points is more significant, leading to overfitting. Hence in the EG-SMOTE algorithm, an upper limit was set for sampling every other minority sub-clusters, as expressed in (2). The algorithm sets up a maximum value for several synthetic samples per each subcluster, where the number of subclusters is decided after Elbow testing. In cluster analysis, the elbow method is a heuristic used in determining the number of clusters in a data set. The Elbow- method consists of plotting the explained variation as a function of the number of clusters and picking the elbow of the curve as the number of clusters to identify the optimal number of clusters. For safe samples, while applying EG-SMOTE to prevent intra-clustering imbalance, if resampling is applied without identifying the optimal number of subclusters, there are possibilities for one sub-cluster to be given less representation. This will again introduce an intra-cluster imbalance than addressing it. Fig.5. represents experiments done with BoT-IoT dataset and results do explain that all classifiers tend to perform well in their optimal cluster number than any other cluster number manually chosen.

Algorithm 2

```
Parameters:
Nodemap $\Phi$ is the Dictionary of assigned labels to nodes.
$L_i$   -   List of all assigned labels to a node
$\Gamma_i$   -   List of modes and i runs from 1 to n
$\gamma_i$   -   One of the modes of labels of the node and
          $\Gamma_i$ is the set of all modes in that node
X   -   Inputs corresponding to all the labels
          assigned to a node
$\Phi_i$   -   ith node
$W_i$   -   weights of ith node
$Y_i$   -   Label of ith dataset
BMN   -   Best Matching Node (winner node)
Start
1. Growing _Self_Organizing_Map ()
   $\Phi = \{\text{‘0:0’:}L_1, \text{‘0:1’: } L_2 \ldots\ldots\text{’x:y’: } L_n\}$
2. Finalize_labels ($\varphi_i$):
   $\Gamma i =$ mode $(l_1, l_2, l_1, l_1 \ldots)$ and $\gamma_i$ ( $\gamma_i \in \Gamma_i$ )
   • If $\exists! \gamma_i$ ($\gamma_i \in \Gamma_i$)
         Update_node_label($\varphi_i$, $\gamma_i$)
   • If $n(\Gamma i) > 1$.
         $di = ||x_i(t) - w||^2; (x \in X)$        (3)
         $BMN(t) = argmini\{d_i\}$        (4)
         $\gamma_i <= Y_i(x_i(t))$
         Update_node_label($\varphi_i$, $\gamma_i$)
3. End
```

B. GSOM ALGORITHM BASED ON HD COMPUTING

As noted earlier, the GSOM algorithm based on HD computing has been demonstrated to be effective in low energy settings for unsupervised learning from unlabeled data. The topological mapping of the GSOM algorithm encapsulates both original and synthesized samples into a structure that can be utilized for threat detection.

The workings of the GSOM algorithm are deliberated as follows. It consists of two phases: first, the growing phase in which the unsupervised learning process grows new nodes and adjusts the neuronal weights to accurately reflect the input space; and the second phase which is the smoothing phase in which the weights are finely adjusted and calibrated to for generalized learning across the input space.

Algorithm 2 is proposed as a post-processing step to GSOM algorithm [40] where a majority-voting label is assigned to each node, unlike the learning phase where multiple labels are associated with each input sample $x(t)$.

Two important steps are defined by the algorithm following the execution of GSOM.

Step 1: The step involves assigning the node with the mode of the labels. This is shown as the transition from subfigure A to B (Fig.6). The imbalance of anomalies in the input samples is mitigated by incorporating EG-SMOTE which synthesized more anomalous samples, thus balancing the data. This will ensure the nodes that represent the anomalies will not be ignored.

Step 2: Given that X contains all the input samples whose labels are associated with a particular node. The second step involves a tie breaker by assigning the label associated with the input x ($x \in$ X) which has the minimal distance to that neuron as calculated using (3) and (4). This is shown by the transition from the subfigure B to C (Fig.6.), assuming that the label associated with the closest inputs for nodes B and D are 1 and 0 respectively.

Algorithm 3

```
Parameters:
W     -    Weights of all nodes in node map
w_i   -    Weights of ith node
Start
    1. $\alpha_i = ||x(t) - w_i||^2; (w \in W),$          (5)
    2. $BMN(t) = argmini\{\alpha_i\},$                   (6)
    3. $\gamma_i <= Y_i(BMN(t))$
    4. Return($\gamma$i)
End
```

C. CLASSIFICATION

After finalizing of the labels in all the nodes, classification will be carried out for each of the new unknown inputs $x1(t)$, as depicted in Algorithm 3.

When there is new data to be classified as whether it is normal or an anomaly, the distance is calculated between the weights of the input vector and the weights of each node. Thereby BMN is determined as the node with the minimum distance by using equations (5) and (6). Then the label associated with that particular node will be given as the prediction for that input sample. For instance, consider the right subfigure C of Fig.6, suppose that node A is the one with the minimum distance to that new unseen input, then the predicted value is 1. This process is repeated until all the values are predicted for the entire test dataset.

IV. EXPERIMENTS

This section provides the details of the results of the experimentation that was conducted to evaluate the EG-SMOTE in handling imbalanced data. We have compared the performance of EG-SMOTE with SMOTE and G-SMOTE across all datasets for the following classifiers, Logistic Regressor (LR), Gradient Boosting Classifier (GBC), K-Nearest Neighbours (KNN), Decision Tree (DT), XGBoost and GSOM classifier. A variety of hyper-parameters were selected based on grid-search and k-fold cross-validation for result comparison and optimization.

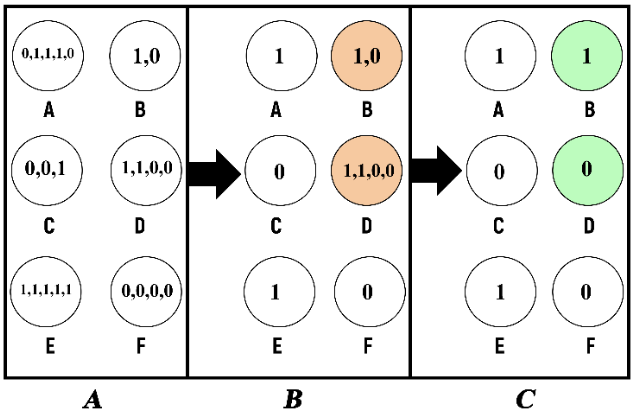

FIGURE 6. Nodes under self-organizing map.

We have evaluated the performance of the classifiers and the oversampling techniques using k-fold cross validation with k = 5. In order to solve the data imbalance in the training set, we applied the oversampling techniques in the k-1 folds of the k-fold cross validation procedure to generate synthetic data and to obtain a balanced training set. The models which are trained on this data are validated on the remaining fold along with performance evaluations. We have tried a number of different hyperparameters for the over samplers and the classifiers. For SMOTE, we have used k $\in$ {5, 3} for the parameter k nearest neighbors, for GSMOTE we have used nearest neighbors k $\in \{5, 3\}$, the deformation factor α def $\in \{1.0, 0.8, 0.6, 0.5, 0.4, 0.2, 0.0\}$ and truncation factor α trunc $\in \{1.0, 0.75, 0.5, 0.25, 0.0, -0.5, -1.0\}$ and finally for EG-SMOTE we have used the same set of hyper parameters that were used for G-SMOTE.

Then for the hyperparameters of the GBC we have tried max depth $\in \{6, 3\}$, learning rate $\in \{0.1, 0.01, 0.001\}$ and the number of estimators $\in \{100, 50\}$, and for the KNN we tried the number of nearest neighbors k $\in \{5, 3\}$ and p $\in \{2, 1\}$, and for the DT we tried max depth $\in \{6, 3\}$ and for the XGBoost we have tried number of estimators $\in$ {800, 600, 400, 200, 100, 50}, learning rate $\in \{0.1, 0.01, 0.001\}$, max depth $\in$ {7, 6, 5, 4, 3}, min_child_weight $\in$ {5, 3, 1}, gamma $\in$ {5, 2, 1.5, 1, 0.5} and col_sample_by_tree $\in \{1.0, 0.8, 0.6\}$. From the above hyperparameters, we have collected the highest evaluation metrics from the cross validation for different over samplers and classifiers for each dataset. We have carried out the experiments 3 times and reported the average values of the results obtained from the experiments.

A. DATASETS

The proposed approach was empirically evaluated using three benchmark datasets: KDD99 [54], NSL-KDD [55], CICIDS2017 [56] and Bot-Iot Dataset [57]. All four datasets exhibit the challenges of imbalanced or skewed data sampling as well as unlabeled data streams in an IoT Edge setting.

TABLE 1. Details of datasets.

Dataset	*Number of attributes*	*Number of Minority*	*Number of Majorities*	*IR*
NSL-KDD	42	5166	73251	14.179
KDD99	41	967	9818	10.153
CICIDS2017	77	1038	43960	42.35
CPES dataset	128	1242	4405	3.546
Bot-Iot Dataset	32	3500	70000	20

Datasets with more than two classes were modified to represent binary classes, and the datasets were pruned to reduce dimensionality after feature ranking. Table 1 shows the details of each dataset (IR represents the imbalanced ratio). Various performance metrics can be used to evaluate a model: F-Score, g-mean, and Area Under the ROC Curve (AUC).

- **F-Score:** Harmonic means of precision and recall and, therefore, balances a model in terms of precision and recall.

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP}), \tag{7}$$

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN}), \tag{8}$$

$$\text{F-Score} = 2^{*}((\text{Precision}^{*}\text{Recall}))/((\text{Precision} +\text{Recall})), \tag{9}$$

- **Area Under the ROC Curve (AUC):** ROC curve results from varying the decision threshold and plotting the true positive rate against the false positive rate.
- **G-mean:** Defined as the geometric mean of Sensitivity and Specificity.

$$\text{Sensitivity} = \text{TP}/(\text{TP}+\text{FN}), \tag{10}$$

$$\text{Specificity} = \text{TN}/(\text{TN}+\text{FP}), \tag{11}$$

$$\text{G-mean} = \sqrt{}(\text{TP}/(\text{TP}+\text{FN})+\text{TN}/(\text{FP}+\text{TN})), \tag{12}$$

TABLE 2. Results for NSL-KDD dataset.

Classifiers	Scores	EG-SMOTE	G-SMOTE	SMOTE
LR	F-Score	0.9600	0.9428	0.9331
	g-mean	0.9800	0.9714	0.9635
	AUC	0.9803	0.9718	0.9640
GBC	F-Score	0.9744	0.9532	0.9566
	g-mean	0.9792	0.9586	0.9704
	AUC	0.9795	0.9595	0.9710
KNN	F-Score	0.9455	0.9575	0.9528
	g-mean	0.9576	0.9774	0.9769
	AUC	0.9588	0.9778	0.9773
DT	F-Score	0.9539	0.9331	0.9377
	g-mean	0.9656	0.9658	0.9661
	AUC	0.9663	0.9663	0.9668
XGBoost	F-Score	0.9699	0.9618	0.9623
	g-mean	0.9810	0.9711	0.9779
	AUC	0.9813	0.9715	0.9783
GSOM	F-Score	0.9647	0.9318	0.9195
	g-mean	0.9752	0.9715	0.9596
	AUC	0.9755	0.9717	0.9600

TABLE 3. Results for KDD dataset.

Classifiers	Scores	EG-SMOTE	G-SMOTE	SMOTE
LR	F-Score	0.8872	0.8605	0.8614
	g-mean	0.9109	0.9794	0.9800
	AUC	0.9145	0.9795	0.9801
GBC	F-Score	0.9969	0.9841	0.9984
	g-mean	0.9969	0.9937	0.9989
	AUC	0.9969	0.9937	0.9989
KNN	F-Score	0.9666	0.9545	0.9675
	g-mean	0.9901	0.9944	0.9962
	AUC	0.9901	0.9944	0.9862
DT	F-Score	0.9802	0.9601	0.9706
	g-mean	0.9858	0.9817	0.9990
	AUC	0.9860	0.9820	0.9990
XGBoost	F-Score	0.9903	0.9734	0.9829
	g-mean	0.9990	0.9973	0.9983
	AUC	0.9990	0.9975	0.9983
GSOM	F-Score	0.9360	0.9465	0.9258
	g-mean	0.9674	0.9584	0.9401
	AUC	0.9677	0.9589	0.9423

TABLE 4. Results for CICID dataset.

Classifiers	Scores	EG-SMOTE	G-SMOTE	SMOTE
LR	F-Score	0.8736	0.8446	0.8506
	g-mean	0.9662	0.9877	0.9879
	AUC	0.9667	0.9877	0.9879
GBC	F-Score	0.9919	0.9815	0.9873
	g-mean	0.9989	0.9912	0.9969
	AUC	0.9989	0.9912	0.9957
KNN	F-Score	0.9948	0.9746	0.9816
	g-mean	0.9952	0.9881	0.9986
	AUC	0.9952	0.9870	0.9986
DT	F-Score	0.9833	0.9348	0.9689
	g-mean	0.9949	0.9856	0.9926
	AUC	0.9949	0.9857	0.9927
XGBOOST	F-Score	0.9728	0.9671	0.9478
	g-mean	0.9989	0.9987	0.9972
	AUC	0.9989	0.9987	0.9972
GSOM	F-Score	0.9999	0.9912	0.9523
	g-mean	0.9999	0.9985	0.9988
	AUC	0.9999	0.9985	0.9988

TABLE 5. Results for Bot-IoT dataset.

Classifiers	Scores	EG-SMOTE	G-SMOTE	SMOTE
LR	F-Score	0.9015	0.9001	0.9034
	g-mean	0.9965	0.9964	0.9941
	AUC	0.9965	0.9964	0.9960
GBC	F-Score	0.9164	0.8964	0.8987
	g-mean	0.9964	0.9962	0.9963
	AUC	0.9964	0.9962	0.9963
KNN	F-Score	0.9356	0.9274	0.9248
	g-mean	0.9795	0.9863	0.9924
	AUC	0.9797	0.9864	0.9924
DT	F-Score	0.9549	0.9460	0.9260
	g-mean	0.9712	0.9723	0.9623
	AUC	0.9716	0.9726	0.9631
XGBOOST	F-Score	0.9812	0.9600	0.9594
	g-mean	0.9930	0.9830	0.9886
	AUC	0.9931	0.9831	0.9886
GSOM	F-Score	0.9445	0.9238	0.9132
	g-mean	0.9780	0.9578	0.9476
	AUC	0.9781	0.9586	0.9488

B. RESULTS

The following tables, TABLE 2, TABLE 3, and TABLE 4 present the mean cross-validation scores for each combination of over samplers, evaluation metrics, and classifiers for NSL-KDD, KDD99, and CICIDS2017 dataset, respectively.

The above experiments were conducted to demonstrate the performance of the EG-SMOTE sampling approach. The results for EG-SMOTE demonstrate a significant improvement for the prediction of anomalous samples in imbalanced datasets as compared with other resampling techniques such as SMOTE and G-SMOTE.

The F-Score reflects the harmonic mean between precision and recall and is considered as a reliable metric for imbalanced classification tasks. G-SMOTE claims that it outperforms Random oversampling SMOTE and borderline SMOTE [35]. The results presented in Table 2, Table 3, and Table 4 suggest that the proposed approach achieves a higher F-Score for most of the classifiers than other over samplers. The classifier based on GSOM performs considerably well, compared to existing classifiers, which confirms its utility in IoT Edge applications. The experiment conducted with the CICID datasets suggest that EG-SMOTE algorithm outperformed all the compared oversampling methods. In addition, EG-SMOTE performs equally well for the new GSOM classifier. Results from the Bot-IoT dataset are presented in Table 5, here again it can be seen that the proposed machine learning approach performs better than the other techniques.

V. CONCLUSION

In this paper, we proposed a novel machine learning method for effective, efficient and secure cyber threat detection at the IoT Edge. The method was empirically evaluated using three benchmark datasets, KDD99, NSL-KDD, CICIDS2017, and an industry-focused botnet IoT traffic dataset, BoT-IoT. Its effectiveness is demonstrated in addressing the challenge of high volume, high velocity unlabeled data streams generated at the IoT Edge. Its efficiency is based on the GSOM algorithm that utilizes HD computing for sparse distributed feature representation and learning from unlabeled data in low-energy settings such as Edge layers. It is secure as it is boosted by minority resampling of imbalanced data generated by cybersecurity threats and attacks at the IoT Edge. Furthermore, the EG-SMOTE algorithm addresses the challenges of synthesizing noisy minority samples, overfitting due to extreme synthesis of minority samples, and improper synthesis along the borderlines due class imbalanced datasets. The GSOM algorithm transforms high-dimensional data into low-dimensional data while preserving the underlying topology representation of the minority resampling boosted datasets generated by the EG-SMOTE algorithm. The latent representation generated by the GSOM algorithm is effective in detecting cyber-physical attacks of varying origins. As future work, we intend to evaluate the proposed approach on a large-scale IoT Edge application, and second, we intend to explore multi-label classification and a safe zone for point generation based on the k-nearest neighbors than relying on the category to improve the efficiency of cyber threat detection at the IoT Edge.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[2] P. Roy, S. Sarker, M. A. Razzaque, M. M. Hassan, S. A. AlQahtani, G. Aloi, and G. Fortino, "AI-enabled mobile multimedia service instance placement scheme in mobile edge computing," *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107573.

[3] R. Ranjan, M. Villari, H. Shen, O. Rana, and R. Buyya, "Software tools and techniques for fog and edge computing," *Softw., Pract. Exper.*, vol. 50, no. 5, pp. 473–475, May 2020.

[4] S. M. Karunarathne, N. Saxena, and M. K. Khan, "Security and privacy in IoT smart healthcare," *IEEE Internet Comput.*, vol. 25, no. 4, pp. 37–48, Jul. 2021.

[5] Y. Tian, B. Song, T. Ma, A. Al-Dhelaan, and M. Al-Dhelaan, "Bi-tier differential privacy for precise auction-based people-centric IoT service," *IEEE Access*, vol. 9, pp. 55036–55044, 2021.

[6] A. A. Cook, G. Misirli, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.

[7] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, Aug. 2018.

[8] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, Jul. 2019.

[9] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, Oct. 2019.

[10] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf, and Y. A. Bangash, "An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10250–10276, Oct. 2020.

[11] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: State of the art and challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1608–1631, Aug. 2019.

[12] B. Sun, X. Shan, K. Wu, and Y. Xiao, "Anomaly detection based secure in-network aggregation for wireless sensor networks," *IEEE Syst. J.*, vol. 7, no. 1, pp. 13–25, Mar. 2013.

[13] D. Rangwani, D. Sadhukhan, S. Ray, M. K. Khan, and M. Dasgupta, "A robust provable-secure privacy-preserving authentication protocol for industrial Internet of Things," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 3, pp. 1548–1571, May 2021.

[14] C. Xenofontos, I. Zografopoulos, C. Konstantinou, A. Jolfaei, M. K. Khan, and K.-K. R. Choo, "Consumer, commercial and industrial IoT (in) security: Attack taxonomy and case studies," *IEEE Internet Things J.*, early access, May 13, 2021, doi: 10.1109/JIOT.2021.3079916.

[15] D. T. Ramotsoela, G. P. Hancke, and A. M. Abu-Mahfouz, "Attack detection in water distribution systems using machine learning," *Hum.-Centric Comput. Inf. Sci.*, vol. 9, p. 13, Dec. 2019.

[16] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," in *Proc. SIAM Int. Conf. Data Mining*, May 2003, pp. 25–36.

[17] V. Jyothsna, V. R. Prasad, and K. M. Prasad, "A review of anomaly based intrusion detection systems," *Int. J. Comput. Appl.*, vol. 28, no. 7, pp. 26–35, Aug. 2011.

[18] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, Nov. 2017.

[19] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS ONE*, vol. 11, no. 4, Apr. 2016, Art. no. e0152173.

[20] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song, "Anomalous example detection in deep learning: A survey," *IEEE Access*, vol. 8, pp. 132330–132347, 2020.

[21] D. M. Hawkins, *Identification of Outliers*, vol. 11. London, U.K.: Chapman & Hall, 1980.

[22] M. Fahim and A. Sillitti, "Anomaly detection, analysis and prediction techniques in IoT environment: A systematic literature review," *IEEE Access*, vol. 7, pp. 81664–81681, 2019.

[23] W. Han, W. Xiong, Y. Xiao, M. Ellabidy, A. V. Vasilakos, and N. Xiong, ''A class of non-statistical traffic anomaly detection in complex network systems,'' in *Proc. 32nd Int. Conf. Distrib. Comput. Syst. Workshops*, Macau, China, Jun. 2012, pp. 640–646.

[24] K. Sha, T. A. Yang, W. Wei, and S. Davari, ''A survey of edge computing-based designs for IoT security,'' *Digit. Commun. Netw.*, vol. 6, no. 2, pp. 195–202, 2020.

[25] A. Huc and D. Trcek, ''Anomaly detection in IoT networks: From architectures to machine learning transparency,'' *IEEE Access*, vol. 9, pp. 60607–60616, 2021.

[26] Y. Yu and N. Bian, ''An intrusion detection method using few-shot learning,'' *IEEE Access*, vol. 8, pp. 49730–49740, 2020.

[27] K. Xu, W. Pedrycz, and Z. Li, ''Granular computing: An augmented scheme of degranulation through a modified partition matrix,'' *Fuzzy Sets Syst.*, vol. 2021, pp. 1–11, Jun. 2021.

[28] M. Ye, X. Wu, X. Hu, and D. Hu, ''Multi-level rough set reduction for decision rule mining,'' *Int. J. Speech Technol.*, vol. 39, no. 3, pp. 642–658, Oct. 2013.

[29] J. Leng, D. Yan, Q. Liu, K. Xu, J. L. Zhao, R. Shi, L. Wei, D. Zhang, and X. Chen, ''ManuChain: Combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 1, pp. 1–11, Jan. 2019.

[30] J. Leng, S. Ye, M. Zhou, J. L. Zhao, Q. Liu, W. Guo, W. Cao, and L. Fu, ''Blockchain-secured smart manufacturing in industry 4.0: A survey,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 1, pp. 237–252, Jan. 2021.

[31] J. Leng, G. Ruan, P. Jiang, K. Xu, Q. Liu, X. Zhou, and C. Liu, ''Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: A survey,'' *Renew. Sustain. Energy Rev.*, vol. 132, Oct. 2020, Art. no. 110112.

[32] M. Signorini, M. Pontecorvi, W. Kanoun, and R. Di Pietro, ''BAD: A blockchain anomaly detection solution,'' *IEEE Access*, vol. 8, pp. 173481–173490, 2020.

[33] H. He and E. A. Garcia, ''Learning from imbalanced data,'' *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[34] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ''SMOTE: Synthetic minority over-sampling technique,'' *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002.

[35] H. Han, W. Wang, and B. Mao, ''Borderline-SMOTE: A new oversampling method in imbalanced data sets learning,'' in *Proc. Int. Conf. Intell. Comput.* Berlin, Germany: Springer, 2005, pp. 878–887.

[36] G. Douzas and F. Bacao, ''Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE,'' *Inf. Sci.*, vol. 501, pp. 118–135, Oct. 2019.

[37] L. Njilla, L. Pearlstein, X.-W. Wu, A. Lutz, and S. Ezekiel, ''Internet of Things anomaly detection using machine learning,'' in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Washington, DC, USA, Oct. 2019, pp. 1–6.

[38] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, ''A geometric framework for unsupervised anomaly detection,'' in *Applications of Data Mining in Computer Security*, vol. 6, D. Barbará and S. Jajodia, Eds. Boston, MA, USA: Springer, 2002, pp. 77–101.

[39] L. Njilla, L. Pearlstein, X.-W. Wu, A. Lutz, and S. Ezekiel, ''Internet of Things anomaly detection using machine learning,'' in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Washington, DC, USA, Oct. 2019, pp. 1–6.

[40] D. Alahakoon, S. K. Halgamuge, and B. Srinivasan, ''Dynamic self-organizing maps with controlled growth for knowledge discovery,'' *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 601–614, May 2000.

[41] T. M. Nam, P. H. Phong, T. D. Khoa, T. T. Huong, P. N. Nam, N. H. Thanh, L. X. Thang, P. A. Tuan, L. Q. Dung, and V. D. Loi, ''Self-organizing map-based approaches in DDoS flooding detection using SDN,'' in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Chiang Mai, Thailand, Jan. 2018, pp. 249–254.

[42] R. Nawaratne, D. Alahakoon, D. De Silva, H. Kumara, and X. Yu, ''Hierarchical two-stream growing self-organizing maps with transience for human activity recognition,'' *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7756–7764, Dec. 2020.

[43] P. Kanerva, ''Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors,'' *Cognit. Comput.*, vol. 1, no. 2, pp. 139–159, Jun. 2009.

[44] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart, ''Distributed representations,'' in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, MA, USA: MIT Press, 1986, pp. 77–109.

[45] D. Kleyko, E. Osipov, D. De Silva, U. Wiklund, V. Vyatkin, and D. Alahakoon, ''Distributed representation of n-gram statistics for boosting self-organizing maps with hyperdimensional computing,'' in *Proc. Int. Andrei Ershov Memorial Conf. Perspect. Syst. Inform.* Cham, Switzerland: Springer, 2019, p. 6479.

[46] D. Kleyko, E. Osipov, D. D. Silva, U. Wiklund, and D. Alahakoon, ''Integer self-organizing maps for digital hardware,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, Jul. 2019, pp. 1–8.

[47] T. Bandaragoda, D. De Silva, D. Kleyko, E. Osipov, U. Wiklund, and D. Alahakoon, ''Trajectory clustering of road traffic in urban environments using incremental machine learning in combination with hyperdimensional computing,'' in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1664–1670.

[48] Y. Wu, ''Cloud-edge orchestration for the Internet of Things: Architecture and AI-powered data processing,'' *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12792–12805, Aug. 2021.

[49] I. Sittón-Candanedo, R. S. Alonso, J. M. Corchado, S. Rodríguez-González, and R. Casado-Vara, ''A review of edge computing reference architectures and a new global edge proposal,'' *Future Gener. Comput. Syst.*, vol. 99, pp. 278–294, Oct. 2019.

[50] V. Q. Rufino, M. S. Nogueira, A. Avritzer, D. S. Menasche, B. Russo, A. Janes, V. Ferme, A. Van Hoorn, H. Schulz, and C. Lima, ''Improving predictability of user-affecting metrics to support anomaly detection in cloud services,'' *IEEE Access*, vol. 8, pp. 198152–198167, 2020.

[51] P. Bholowalia and A. Kumar, ''EBK-Means: A clustering technique based on elbow method and K-means in WSN,'' *Int. J. Comput. Appl*, vol. 105, no. 9, pp. 17–24, 2014.

[52] M. Perez-Ortiz, P. A. Gutierrez, P. Tino, and C. Hervas-Martinez, ''Oversampling the minority class in the feature space,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1947–1961, Sep. 2016.

[53] B. Krawczyk, ''Learning from imbalanced data: Open challenges and future directions,'' *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016.

[54] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, ''Cost-based modeling for fraud and intrusion detection: Results from the JAM project,'' in *Proc. DARPA Inf. Survivability Conf. Expo. (DISCEX)*, Hilton Head, SC, USA, vol. 2, Jan. 1999, pp. 130–144.

[55] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, ''A detailed analysis of the KDD CUP 99 data set,'' in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Ottawa, ON, Canada, Jul. 2009, pp. 1–6.

[56] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ''Toward generating a new intrusion detection dataset and intrusion traffic characterization,'' in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, Madeira, Portugal, 2018, pp. 108–116.

[57] 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 Gener. Comput. Syst.*, vol. 100, pp. 779–796, 2019.

VIVEK CHRISTOPHER is currently pursuing the bachelor's degree in computer science and engineering with the University of Moratuwa. He is also employed at WSO2 and continuing research and development in IAM domain. He has been involved in deep learning and machine learning-based research projects for his final year at the university. He has done his internship at WSO2 Lanka (Private) Ltd. His research interests include affective computing, security systems, and IAM.

THARMASANTHIRAN AATHMAN is currently pursuing the bachelor's degree in computer science and engineering with the University of Moratuwa. He is also employed at WSO2 pursuing research and development in artificial intelligence and performance analysis. He has been involved in deep learning and machine learning-based research projects for his final year at the university. He has done his internship at WSO2 Lanka (Private) Ltd. His research interests include machine learning, deep learning, and computer vision.

KAYATHIRI MAHENDRAKUMARAN is currently pursuing the bachelor's degree with the Department of Computer Science and Engineering, University of Moratuwa. She is also working as a Software Engineer at WSO2 and continuing the research work in IAM domain. She has done her internship at WSO2 Lanka (Private) Ltd. She has done projects related to machine learning and deep learning during her final year at the University. Her research interests include machine learning, deep learning, and IAM.

RASHMIKA NAWARATNE received the B.S. degree (Hons.) from the Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka, in 2014. He is currently pursuing the Ph.D. degree in data analytics and cognition with La Trobe University, Australia. Prior to commencing his Ph.D., he was employed at a software product development organization in the capacity of a Technical Lead. His research interests include self-learning, incremental learning, video analytics, deep learning, and human cognition.

DASWIN DE SILVA (Senior Member, IEEE) received the Ph.D. degree in artificial intelligence from Monash University, Australia. He is currently an Associate Professor and the Deputy Director of the Centre for Data Analytics and Cognition, La Trobe University, Australia. His research interests are diverse and include autonomous learning, active perception, information fusion, cognitive computing, neuromorphic computing, natural language processing, deep emotions, psycholinguistics, and intelligent cloud platforms. He is also an Associate Editor of the IEEE Transactions on Industrial Informatics, the IEEE Open Journal of the Industrial Electronics Society, and *Discover AI*.

VISHAKA NANAYAKKARA received the B.Sc. (Eng.) degree specializing in computer science and engineering from the University of Moratuwa, Sri Lanka, in 1994, and the Technical Licentiate degree in computer engineering from Chalmers University of Technology, Sweden, in 2002.

She joined the teaching faculty of the university upon graduation and was appointed the Head of the Department of Computer Science and Engineering, in 2005, and served in that capacity for six years. During her sabbatical leave, in December 2011, she worked as the Dean of the Faculty of Electrical and Information Technology, Northshore College of Business and Technology. She was then appointed as the Deputy Project Director of the Higher Education for the Twenty First Century (HETC) Project of the Ministry of Higher Education of Sri Lanka, from 2013 to 2015. She is currently a Senior Lecturer. She is also the Director of the Centre for Open and Distance Learning, University of Moratuwa. She has been instrumental in setting up the DataSERACH—multi-disciplinary research center engaged in research in data science, engineering, and analytics at the University of Moratuwa and setting up of the new data science and engineering stream at the Department of Computer Science and Engineering.

Mrs. Nanayakkara serves as the Board Director of the Women's Chamber for Digital Sri Lanka and LIRNEasia. In 2016, she was awarded the ''Female ICT Leader of the Year'' by the Computer Society of Sri Lanka.

DAMMINDA ALAHAKOON (Member, IEEE) received the Ph.D. degree in artificial intelligence from Monash University, Australia. He is currently a Full Professor and the Founding Director of the Centre for Data Analytics and Cognition, La Trobe University, Australia. He has made significant contributions with international impact toward the advancement of artificial intelligence through academic research, applied research, research supervision, industry engagement, curriculum development, and teaching. He has published over 100 research articles; theoretical research in self-structuring AI, human-centric AI, cognitive computing, deep learning, optimization; and applied AI research in industrial informatics, smart cities, robotics, intelligent transport, digital health, energy, sport science, and education.